

BAB II

LANDASAN TEORI

II.1. Sistem

II.1.1. Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu sekelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya. Pendekatan yang menekankan pada prosedur dan mendefinisikan sistem sebagai suatu jaringan kerja prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan sistem sebagai kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. kedua kelompok definisi ini adalah benar dan tidak bertentangan. Yang berbeda adalah cara pendekatannya (Tata Sutabri ; 2012 : 2).

Teori sistem melahirkan konsep-konsep futuristik, antara lain yang terkenal adalah konsep sibernetika (*cyberbetics*). Konsep atau bidang kajian ilmiah ini terutama berkaitan dengan upaya-upaya untuk menerapkan berbagai disiplin ilmu, yaitu ilmu perilaku, fisikan, biologi, dan teknik. Oleh karena itu sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan oleh manusia, sehingga melahirkan studi-studi tentang robotika, kecerdasan buatan (*artificial intelegence*) dan lain sebagainya. Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolaha

(*processing*) dan keluaran (*output*). Di samping itu suatu sistem senantiasa tidak terlepas dari lingkungan sekitarnya. Maka umpan balik (*feed-back*) selain dapat berasal dari *output*, juga dapat berasal dari lingkungan sistem tersebut. Organisasi dipandang sebagai suatu sistem yang juga memiliki semua unsur (Tata Sutabri ; 2012 : 3).

II.1.2. Karakteristik Sistem

Model umum sebuah sistem terdiri dari *input*, proses dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem.

Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya.

Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem tersebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh di dalam suatu unit sistem komputer,

“program” adalah *maintenance input* yang digunakan untuk mengoperasikan *computer*. Sementara “data” adalah *signal* yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, di mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan *input* bagi subsistem lainnya.

7. Pengolahan Sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi, sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan (Tata Sutabri ; 2012 : 13-14).

II.1.3. Pengertian Sistem

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling bergantung satu sama lain dan terpadu (Tata Sutabri ; 2012 : 3).

Gordon B. Davis dalam bukunya menyatakan bahwa sistem bisa berupa abstrak atau fisik. Sistem yang abstrak adalah susunan gagasan-gagasan atau konsepsi yang teratur yang saling bergantung. Sedangkan sistem yang bersifat fisik adalah serangkaian unsur yang bekerja sama untuk mencapai suatu tujuan (Tata Sutabri ; 2012 : 6).

Sedangkan Norman L. Enger menyatakan bahwa suatu sistem dapat terdiri atas kegiatan-kegiatan yang berhubungan guna mencapai tujuan-tujuan perusahaan seperti pengendalian inventaris atau penjadwalan produksi (Tata Sutabri ; 2012 : 7).

II.2. Sistem Pendukung Keputusan (*Decision Support System*)

Sistem pendukung keputusan merupakan suatu penerapan sistem informasi yang ditujukan untuk membantu pimpinan dalam proses pengambilan keputusan. Sistem pendukung keputusan menggabungkan kemampuan komputer dalam pelayanan interaktif dengan pengolahan atau manipulasi data yang memanfaatkan model atau aturan penyelesaian yang tidak terstruktur (Eva Yulianti ; 2015 : 55)

Sistem pendukung keputusan/ *Decision Support Sistem (DSS)* termasuk bagian dari sistem informasi berdasarkan dukungan kepada pemakainya, dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi terstruktur yang spesifik. Sebelum mengetahui definisi sistem pendukung keputusan alangkah baiknya mengetahui konsep sistem pendukung keputusan terlebih dahulu, mulai dari sistem, informasi, keputusan dan sistem informasi (Elena Monica, dkk ; 2015 : 63).

Menurut Herbert A.Simon, “Sistem Pendukung Keputusan merupakan pengabungan sumber-sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem pendukung keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah-masalah yang struktur”. Dengan pengertian diatas dapat dijelaskan bahwa sistem pendukung keputusan bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem membantu pengambil keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan dengan lebih cepat dan akurat. Sehingga sistem ini tidak dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan (Rika Yunitarini ; 2013 : 45-46).

II.2.1. Karakteristik Sistem Pendukung Keputusan

Karakteristik dalam sistem pendukung keputusan, antara lain :

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 modelmodel analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari / interogasi informasi.
3. Sistem Pendukung Keputusan, dirancang sedemikianrupa sehingga dapat digunakan/dioperasikan dengan mudah.
4. Sistem Pendukung Keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi (Erliza Septia Nagara, dkk ; 2015 : 4).

II.3. Metode SMART (*Simple Multi-Attribute Rating Technique*)

SMART (Simple Multi – Atribut Technique) adalah suatu metode yang digunakan untuk mencari alternatif dari sejumlah alternatif optimal dengan kriteria tertentu. Inti dari *SMART* adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perangkingan yang akan menyeleksi alternatif yang sudah diberikan. *SMART* menggunakan perkalian untuk menghubungkan nilai atribut (kriteria), dimana nilai setiap atribut (kriteria) harus dipangkatkan dulu dengan bobot (kriteria) yang bersangkutan (Mukhsin Nasution ; 2014 : 13).

Metode pembobotan *SMART* merupakan metode pendukung keputusan yang paling sederhana. Dalam metode ini dilihat beberapa parameter yang menjadi penentu keputusan tersebut, Parameter tersebut mempunyai *range* nilai dan bobot yang berbeda-beda. Nilai tersebut nantinya akan menjadi penentu keputusan yang diambil (Eva Yulianti ; 2015 : 56).

SMART merupakan metode dalam pengambilan keputusan multiatribut. Teknik pengambilan keputusan multiatribut ini digunakan untuk mendukung pembuat keputusan dalam memilih beberapa alternatif. Setiap pembuat keputusan harus memiliki sebuah alternatif yang sesuai dengan tujuan yang dirumuskan. Setiap alternatif terdiri dari sekumpulan atribut dan setiap atribut mempunyai nilai-nilai. Nilai ini dirata-rata dengan skala tertentu. Setiap atribut mempunyai bobot yang menggambarkan seberapa penting suatu atribut dibandingkan dengan atribut lain. Pembobotan dan pemberian peringkat ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik (Rika Yunitarini ; 2013 : 46).

II.3.1. Langkah-Langkah Metode *SMART*

Adapun langkah-langkah dalam proses perhitungan metode *SMART* dapat ditunjukkan sebagai berikut (Yeni Kustiyahningsih, dkk ; 2012 : 22-23).

1. Langkah 1: menentukan jumlah kriteria.
2. Langkah 2: sistem secara *default* memberikan skala 0-100 berdasarkan prioritas yang telah diinputkan kemudian dilakukan normalisasi.

$$\text{Normalisasi} = \frac{w_j}{\sum w_j} \quad (1)$$

Keterangan : w_j : bobot suatu kriteria

$\sum w_j$: total bobot semua kriteria

3. Langkah 3: memberikan nilai kriteria untuk setiap alternatif.
4. Langkah 4: hitung nilai *utility* untuk setiap kriteria masing-masing.

$$u_i(a_i) = 100 \frac{(C_{out\ i} - C_{min})}{(C_{max} - C_{min})} \% \quad (2)$$

Keterangan :

$u_i(a_i)$: nilai *utility* kriteria ke-1 untuk kriteria ke-i

C_{max} : nilai kriteria maksimal

C_{min} : nilai kriteria minimal

$C_{out\ i}$: nilai kriteria ke-i

5. Langkah 5: hitung nilai akhir masing-masing.

$$u(a_i) = \sum_{j=1}^m w_j u_i(a_i), \quad (3)$$

II.4. Pengertian Database

Sebuah basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi. Di dalam basis data, semua data diintegrasikan dengan departemen dan pemakai. Basis data tidak hanya memegang data operasional organisasi, tetapi juga penjelasan mengenai data tersebut. Karena alasan tersebut basis data dapat juga dideskripsikan sebagai kumpulan data yang saling terintegrasi. Basis data juga merupakan sekumpulan elemen data terintegrasi yang secara logika saling berhubungan. Basis data

mengonsolidasikan berbagai catatan yang terlebih dahulu disimpan dalam *file-file* terpisah ke dalam satu gabungan umum elemen data yang menyediakan data untuk banyak aplikasi. Elemen data mendeskripsikan entitas-entitas dan hubungan antara entitas-entitas tersebut (Indrajani ; 2015 : 70).

II.5. Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi. Kamus data terdapat pada tahapan analisis dan perancangan. Pada tahap analisis, kamus data berfungsi untuk mendefinisikan data yang mengalir pada sistem. Sedangkan pada tahap perancangan, kamus data ini digunakan untuk merancang masukan dan keluaran seperti laporan serta basis data. Pada DFD aliran data memiliki sifat global, sedangkan pada kamus data dibuat berdasarkan aliran data yang terdapat pada DFD (Indrajani ; 2015 : 30).

Sumber kamus data, yaitu:

1. *Data Store (File-file)*
2. *Data Flow (Aliran data)*
3. *Data element* yang dinyatakan dalam spesifikasi data dan berasal dari *file*.

Tabel II.1. Notasi Kamus Data

Notasi	Keterangan
=	<i>Is Composed Of</i>
+	<i>And</i>
()	<i>Optional (May be present or absent)</i>
{ }	<i>Iteration</i>

[]	Select one of several alternative choices
**	Comment
@	Identifier (key field) for a store
	Separates alternative choices in the construct

(Sumber : Indrajani ; 2015 : 31)

Contoh kamus data, antara lain:

name = *courtesy-title* + *first-name* +(middle-name) + *last-name*

courtesy-title = [Mr. | Miss | Mrs. | Ms. | Dr. | Profesor]

first-name = {legal-character}

middle-name = {legal-character}

last-name = {legal-character}

legal-character = [A-Z|a-z|0-9|'|_|]

II.6. Normalisasi

Normalisasi adalah suatu teknik dengan pendekatan *bottom-up* yang digunakan untuk membantu mengidentifikasi hubungan. Dimulai dari menguji hubungan, yaitu *functional dependencies* antara atribut. Pengertian lainnya adalah suatu teknik yang menghasilkan sekumpulan hubungan dengan sifat-sifat yang diinginkan dan memenuhi kebutuhan pada perusahaan (Indrajani ; 2015 : 7).

Proses normalisasi merupakan proses pengelompokan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Proses ini selalu diuji pada beberapa kondisi. Apakah ada kesulitan pada saat menambah (*insert*), menghapus (*delete*), mengubah (*update*), atau membaca (*retrieve*) pada suatu

database. Bila ada kesulitan pada pengujian tersebut maka relasi dapat dipecah dalam beberapa tabel lagi (Tata Sutabri ; 2012 : 138).

Adapun bentuk-bentuk normalisasi menurut (Indrajani ; 2015 : 9-10) adalah sebagai berikut:

1. *Unnormalized Form* (UNF)

Merupakan suatu tabel yang berisikan satu atau lebih grup yang berulang. Membuat tabel yang *unnormalized*, yaitu dengan memindahkan data dari sumber informasi.

Contoh: nota penjualan yang disimpan ke dalam format tabel dengan baris dan kolom.

2. *First Normal Form* (1NF)

Merupakan sebuah relasi di mana setiap baris dan kolom berisikan satu dan hanya satu nilai.

Proses UNF ke 1NF

- a. Tentukan satu atau sekumpulan atribut sebagai kunci untuk tabel *unnormalized*.
- b. Identifikasikan grup yang berulang dalam tabel *unnormalized* yang berulang untuk kunci atribut.

3. *Second Normal Form* (2NF)

Berdasarkan pada konsep *full functional dependency*, yaitu A dan B merupakan atribut sebuah relasi. B dikatakan *fully dependent* terhadap A jika B *functional dependent* pada A tetapi tidak pada propersubset dari A. 2NF merupakan sebuah relasi dalam 1NF dan setiap atribut

non-primary-key bersifat *fully functionally dependent* pada *primary key*.

1NF ke 2NF

- a. Identifikasikan *primary key* untuk relasi 1NF.
- b. Identifikasikan *functional dependencies* dalam relasi.
- c. Jika terdapat *partial dependencies* terhadap *primary key*, maka hapus dengan menempatkan dalam relasi yang baru bersama salinan determinannya.

4. *Third Normal Form (3NF)*

Berdasarkan pada konsep *transitive dependency*, yaitu suatu kondisi di mana A, B, dan C merupakan atribut sebuah relasi, maka A – B dan B – C, maka *transitively dependent* pada A melalui B. 3NF adalah sebuah relasi dalam 1NF dan 2NF, di mana tidak terdapat atribut *non primary key* yang bersifat *transitively dependent* pada *primary key*.

2NF ke 3NF

- a. Identifikasi *primary key* dalam relasi 2NF.
- b. Identifikasi *functional dependencies* dalam relasi.
- c. Jika terdapat *transitive dependencies* terhadap *primary key* hapus dengan menempatkannya dalam relasi yang baru bersama dengan salinan determinannya.

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

Berdasarkan pada *functional dependencies* yang dimasukkan dalam hitungan seluruh *candidate key* dalam suatu relasi. Bagaimanapun

BCNF juga memiliki batasan-batasan tambahan disamakan dengan definisi umum dari 3NF. Suatu relasi dikatakan BCNF, jika dan hanya jika setiap determinan merupakan *candidate key*. Perbedaan antara 3NF dan BCNF yaitu untuk *functional dependent* $A \rightarrow B$, 3NF memungkinkan *dependency* ini dalam suatu relasi jika adalah atribut *primary key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menetapkan dengan jelas bahwa untuk *dependency* ini agar ditetapkan dalam relasi A, maka A harus merupakan *candidate key*. Setiap relasi dalam BCNF juga merupakan 3NF, tetapi relasi dalam 3NF belum tentu termasuk ke dalam BCNF. Dalam BCNF kesalahan jarang sekali terjadi, Kesalahan dapat terjadi pada relasi yang (Indrajani ; 2015 : 9-10):

- a. Terdiri atas 2 atau lebih *composite candidate key*.
- b. *Candidate key overlap*, sedikitnya satu atribut.

II.7. Entity Relationship Diagram (ERD)

Entity Relational (ER) Modeling adalah sebuah pendekatan *top-bottom* dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut dengan entitas dan hubungan antara entitas-entitas tersebut yang digambarkan dalam suatu model. Karena terdapat keterbatasan pada ER Model, maka terdapat pengembangan penambahan konsep semantik pada ER yang disebut *Enhanced Entity Relational* (EER) Model (Indrajani ; 2015 : 17).

Relasi antara dua *file* atau dua tabel dapat dikategorikan menjadi 3 macam. Demikian pula untuk membantu menggambarkan relasi secara lengkap terdapat juga beberapa relasi dalam hubungan atribut yang ada dalam satu atau dua *file* (Tata Sutabri ; 2012 : 144).

Adapun relasi yang terjadi di antara dua himpunan entitas dapat berupa sebagai berikut:

1. *One to one relationship* dua *file*

Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu. Seperti pada pelajaran privat di mana satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru.

2. *One to many relationship* dua *file*

Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak lawan satu. Seperti pada sistem pengajaran di sekolah dasar, di mana satu guru mengajar banyak siswa dan siswa hanya diajar oleh satu guru.

3. *Many to many relationship* dua *file*

Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran di perguruan tinggi, di mana satu dosen mengajar banyak mahasiswa dan mahasiswa diajar oleh banyak dosen (Tata Sutabri ; 2012 : 144-145).

II.7.1. *Flowchart*

Flowchart program adalah suatu bagan yang menggambarkan atau mempresentasikan suatu algoritma atau prosedur untuk menyelesaikan masalah. *Flowchart* terbagi menjadi dua yaitu *flowchart system* dan *flowchart program*.

1. *Flowchart System*

Yaitu bagan yang menggambarkan suatu prosedur dan proses suatu *file* dalam suatu media menjadi suatu *file* dalam media yang lain dalam suatu sistem data.

2. *Flowchart Program*

Yaitu bagan yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah (Elena Monica, dkk ; 2015 : 69).

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

UML adalah metode pemodelan berorientasi objek. Sebelum UML telah banyak dibuat pemodelan berorientasi objek seperti: Simula -67, *Objective C*, C++, Eiffel, dan CLOS. Sesuai dengan namanya, *Unified*, pada dasarnya UML merupakan pemodelan yang generik, Sifatnya yang *general* memudahkan UML untuk digunakan pada berbagai tipe sistem yang dimodelkan.

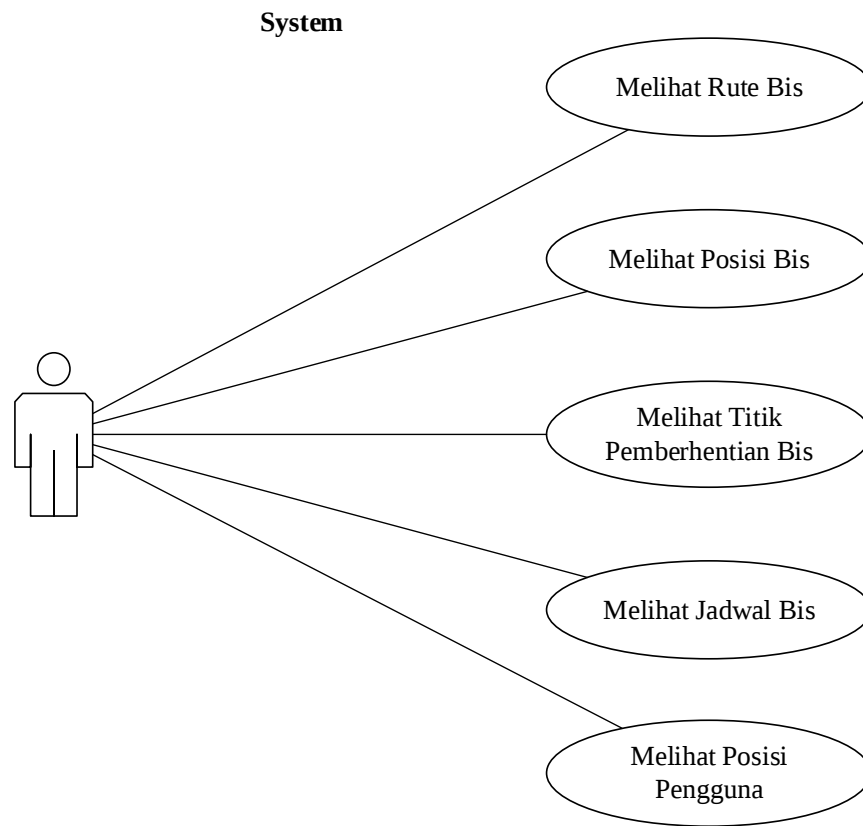
Pada level atas, cara pandang dalam UML dapat dibagi menjadi tiga area: *structural clasification*, *dynamic behavior*, dan model *management*. Klasifikasi struktural menjelaskan suatu objek dalam sistem dan hubungan antar objek tersebut. Pembahasan di dalamnya menyangkut *class*, *use case*, komponen dan *node*. *Dynamic behavior* menjelaskan perilaku sistem dari waktu ke waktu.

Termasuk dalam *dynamic, behavior view* adalah *state machine view, activity view, dan interaction view*. Model *management* menjelaskan pengorganisasian model secara hierarkis (Maman Abdurohman, dkk ; 2010 : 93).

Menurut (Indrajani ; 2015 : 45) diagram-diagram yang termasuk di dalam UML, yaitu:

1. Use Case Diagram

Merupakan suatu diagram yang berisi *use case, actor*, serta *relationship* di antaranya. *Use case* diagram merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan apa saja yang diperlukan dari suatu sistem. Jadi, dapat digambarkan dengan *detail* bagaimana suatu sistem memproses atau melakukan sesuatu, bagaimana cara *actor* akan menggunakan sistem, serta apa saja yang dapat dilakukan terhadap suatu sistem. Notasi yang digunakan dalam *Use Case* adalah persegi panjang yang merupakan *system boundary*, *oval* yang merupakan suatu proses, dan gambar orang yang berinteraksi dalam proses tersebut. Adapun mengilustrasikan *actor, use case* dan *boundary* seperti Gambar II.1. di bawah ini:

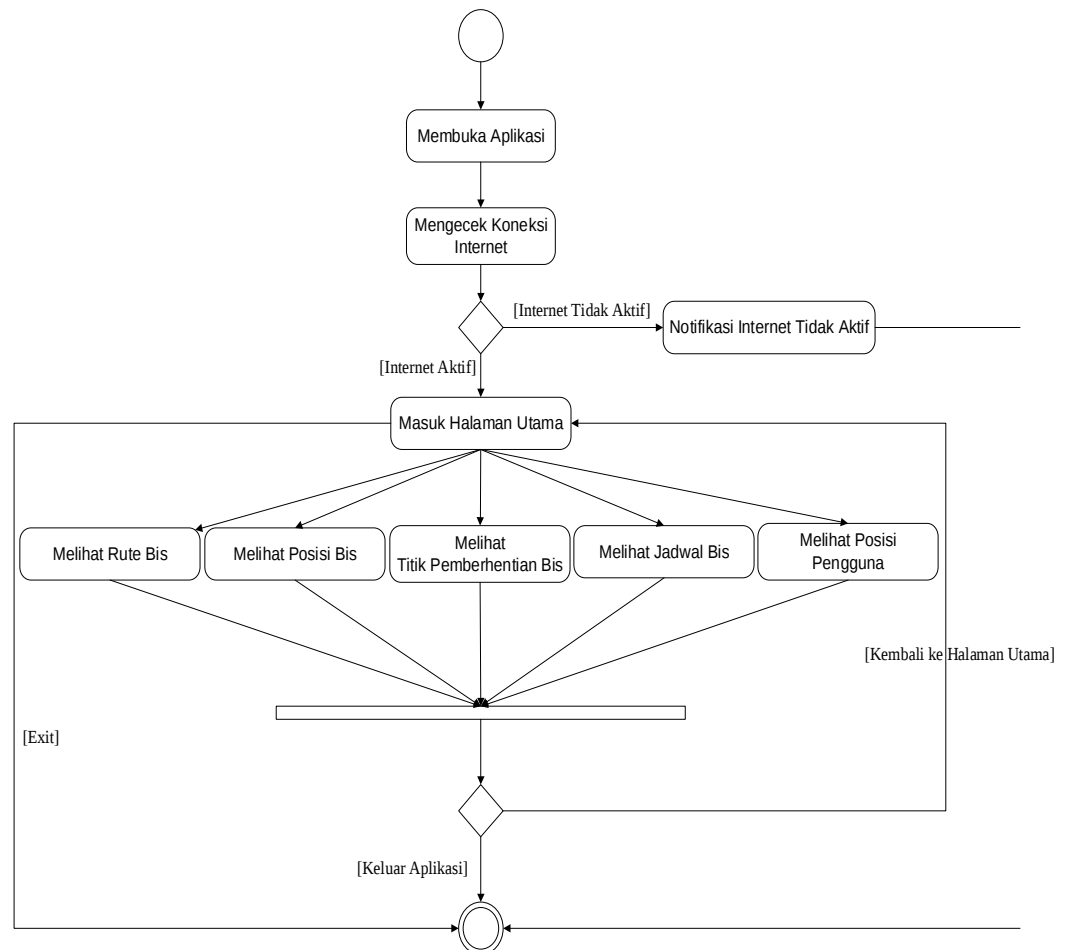


Gambar II.1. Contoh *Use Case Diagram*
 (Sumber : Edgar Winata, dkk ; 2013 : 38)

2. *Activity Diagram*

Digunakan untuk menganalisis *behaviour* dengan *use case* yang lebih kompleks dan menunjukkan interaksi-interaksi di antara mereka satu sama lain. *Activity diagram* sebenarnya memiliki kesamaan dengan *statechart activity diagram* banyak digunakan pada saat peralihan antara analisis dan fase desain. Umumnya digunakan untuk menggambarkan sistem interaktif yang “*real time*”. Diagram ini bersifat optional dalam suatu perancangan sistem. Kumpulan sub-*state*

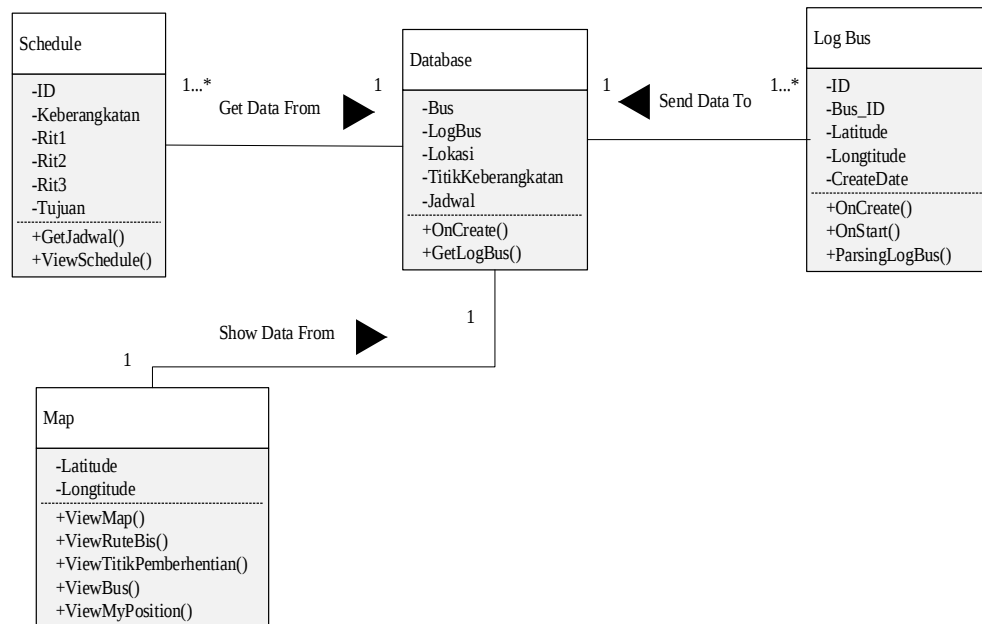
yang dikelompokkan ke dalam sebuah *state* disebut sebagai *composite state*.



Gambar II.2. Contoh Activity Diagram
(Sumber : Edgar Winata, dkk ; 2013 : 39)

3. Class Diagram

Diagram ini digunakan untuk menggambarkan perbedaan yang mendasar antara *class-class*, hubungan antar-*class*, dan di mana sub-sistem *class* tersebut. Pada *class diagram* terdapat nama *class*, *attributes*, *operations*, serta *association* (hubungan antar-*class*).

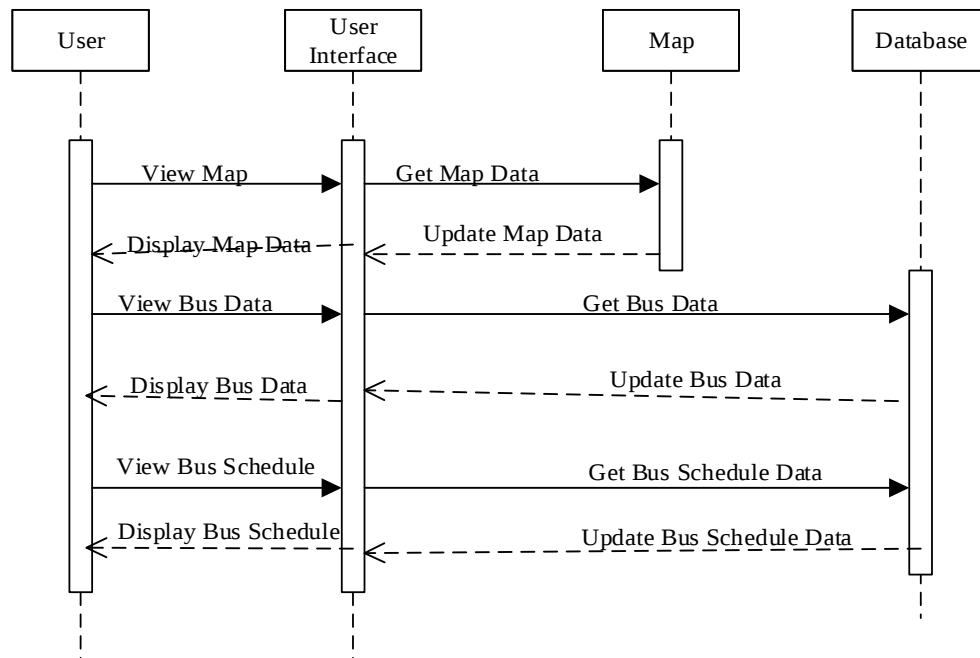


Gambar II.3. Contoh Class Diagram
(Sumber : Edgar Winata, dkk ; 2013 : 38)

4. Sequence Diagram

Merupakan suatu diagram interaksi yang menggambarkan bagaimana objek-objek berpartisipasi dalam bagian interaksi (particular *interaction*) dan pesan yang ditukar dalam urutan waktu. *Sequence* diagram merupakan peralatan untuk interaksi berkomunikasi diagram. Sebuah interaksi didesain antara objek atau sistem yang berpartisipasi dalam sebuah kolaborasi. Beberapa konsep dapat diterapkan pada konteks yang bervariasi. *Sequence* diagram menggambarkan interaksi antara objek secara beraturan sesuai dengan waktu. *Sequence* diagram dapat digambarkan dalam beberapa level secara *detail* dan untuk tujuan yang berbeda pada beberapa langkah yang dikembangkan

secara *lifecycle*. Adapun simbol-simbol *sequence* diagram dapat dilihat Gambar II.4. berikut:



Gambar II.4. Contoh Sequence Diagram
(Sumber : Edgar Winata, dkk ; 2013 : 38)

II.9. Microsoft Visual Basic 2010

Pada akhir tahun 1999, Teknologi *.NET* diumumkan. *Microsoft* memosisikan teknologi tersebut sebagai *platform* untuk membangun *XML Web Services*. *XML Web services* memungkinkan aplikasi tipe manapun dan dapat mengambil data yang tersimpan pada *server* dengan tipe apapun melalui internet (Priyanto Hidayatullah ; 2012 : 4).

Visual Basic.NET adalah *Visual Basic* yang direkayasa kembali untuk digunakan pada *platform .NET* sehingga aplikasi yang dibuat menggunakan *Visual Basic .NET* dapat berjalan pada sistem komputer apa pun, dan dapat

mengambil data dari *server* dengan tipe apa pun asalkan terinstal *.NET Framework* (Priyanto Hidayatullah ; 2012 : 5).

II.10. *Microsoft SQL Server 2008*

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang dipergunakan dalam manajemen basis data relasional. Saat ini hampir semua *server* basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

SQL terdiri dari dua bahasa, yaitu *Data Definition Language Manipulation Language (DDL)*. Implementasi DDL dan DML sistem manajemen basis data (*SMBD*), namun secara umum implemen bahasa ini memiliki bentuk standar yang ditetapkan oleh *ANSI*.

1. *Data Defenition Language (DDL)*

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, *user*, dan sebagainya. *DDL* biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

Secara umum *DDL* yang digunakan adalah:

- a. *CREATE* untuk membuat objek baru.
- b. *USE* untuk menggunakan objek yang sudah ada.
- c. *ALTER* untuk mengubah objek yang sudah ada.
- d. *DROP* untuk menghapus objek.

2. *Data Manipulation Language (DML)*

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah-perintah yang umum dilakukan adalah:

- a. *SELECT* untuk menampilkan data.
- b. *INSERT* untuk menambahkan data baru.
- c. *UPDATE* untuk mengubah data yang sudah ada.
- d. *DELETE* untuk menghapus data (Adelia, Jimmy Setiawan ; 2011 : 115).