

BAB II

LANDASAN TEORI

II.1. Weighted Product

Fuzzy Multiple Attribute Decision Making (FMADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu.

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah FMADM. antara lain :

- a. Simple Additive Weighting Method (SAW)
- b. Weighted Product (WP)
- c. Electre
- d. Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)
- e. Analytic Hierarchy Process (AHP)

WP adalah salah satu metode pengambilan keputusan multi-kriteria. Model ini mirip dengan model jumlah terbobot / weighted sum model (WSM). Perbedaan utama adalah pada WSM dilakukan penambahan bobot dan pada WP dilakukan perkalian bobot. Setiap alternatif keputusan dibandingkan dengan yang lain dengan mengalikan sejumlah rasio, satu untuk setiap kriteria keputusan.

Preferensi untuk alternatif A_i diberikan sebagai berikut:

$$S_i = \prod_{j=1}^n x_{ij}^{w_j}$$

Rumus 1 Metode Weighted Product

dengan $i=1,2,\dots,n$; dimana

$$\sum w_j = 1$$

w_j adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya. (Jurnal Sains dan Terapan Politeknik Hasnur : 2014 : 34-35)

Secara umum, prosedur WPM mengikuti langkah-langkah sebagai berikut:

1. Menentukan kriteria yang akan digunakan sebagai parameter penilaian.
2. Melakukan perhitungan nilai relatif bobot awal (w_j). Nilai bobot awal (w_0) digunakan untuk menunjukkan tingkat kepentingan relatif dari setiap kriteria. Nilai bobot awal (w_0) dinormalisasi menggunakan rumus 1 sehingga total nilai relatif bobot awal $\sum w_j = 1$

$$w_j = \frac{w_0}{\sum w_0} \quad (1)$$

3. Melakukan perhitungan nilai preferensi untuk setiap alternatif A_i (vektor S). Perhitungan nilai preferensi untuk alternatif A_i diawali dengan memberikan nilai rating kinerja perumahan ke- i terhadap kriteria ke j (x_{ij}). Setelah masing-masing kandidat perumahan diberi nilai rating kinerja, nilai ini akan dipangkatkan dengan nilai relatif bobot yang telah dihitung sebelumnya (w_j). w_j akan bernilai positif untuk atribut *benefit* (keuntungan) dan bernilai negatif untuk atribut *cost* (biaya). Rumus yang digunakan untuk menghitung nilai preferensi untuk setiap alternatif A_i (vektor S) adalah rumus 2.

$$S_i = \prod_{j=1}^n (x_{ij})^{w_j} \quad \text{dengan } i = 1,2,3,\dots, m \quad (2)$$

Melakukan perhitungan nilai preferensi relatif dari setiap alternatif menggunakan rumus 3.

$$V_1 = \frac{\prod_{j=1}^n (x_{ij}) w_j}{\prod_{j=1}^n (x_j^*) w_j} \quad \text{dengan } i = 1, 2, 3, \dots, m \quad (3)$$

Perhitungan nilai preferensi untuk alternatif A_i dilakukan dengan membagi nilai vektor S pada perumahan ke- i dan kriteria ke- j , (Rahmadi Wijaya::66-67)

II.1.1. Sistem

Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*). (Kusrini; 2007 : 11).

II.1.2. Informasi

Informasi merupakan hasil olah data, dimana data tersebut sudah diproses dan diinterpretasikan menjadi sesuatu yang bermakna untuk mengambil keputusan. Informasi juga diartikan sebagai himpunan dari data yang relevan dengan satu atau beberapa orang dalam suatu waktu. Suatu informasi berguna bagi pembuat keputusan karena informasi bisa menurunkan ketidakpastian (meningkatkan pengetahuan) tentang hal yang sedang dipikirkan. Makna dari sebuah informasi tentu berbeda-beda diantara seorang dengan lainnya. Tergantung pada tingkat kepentingannya. (Kusrini; 2007 : 4).

Kegunaan informasi bagi seseorang tergantung pada waktu.pada suatu waktu tertentu informasi tersebut mungkin sangat diperlukan di lain hari, mungkin

saja hal tersebut sudah tidak berguna sama sekali. Contohnya, informasi perbandingan harga barang akan sangat dibutuhkan oleh seseorang yang akan membeli barang tersebut. Namun saat ini dia sedang tidak mempertimbangkan untuk membeli barang tersebut, informasi tersebut menjadi kurang bermakna. (Kusrini; 2007 : 4).

II.1.3. Pengertian Sistem Informasi

Suatu sistem di dalam 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. (Kusrini; 2007 : 11).

II.1.4. Sistem Pendukung Keputusan

DSS merupakan system informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. (Kusrini; 2007 : 15).

DSS lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan criteria yang kurang jelas. (Kusrini; 2007 : 16).

II.1.5. Arsitektur Sistem Pendukung Keputusan

Aplikasi sistem pendukung keputusan bisa terdiri dari beberapa subsistem, yaitu :

1. Subsistem manajemen data

Subsistem manajemen data memasukkan satu database yang berisi data yang relevan untuk suatu situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen database (*DBMS / Data Base Management System*). Subsistem manajemen data bisa diinterkoneksi dengan data *warehouse* perusahaan, suatu repositori untuk data perusahaan yang *relevan* dengan pengambilan keputusan. (Kusrini; 2007 : 25).

2. Subsistem manajemen model

Merupakan paket perangkat lunak yang memasukkan model keuangan, statistik, ilmu manajemen, 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. (Kusrini; 2007 : 25).

3. Subsistem antarmuka pengguna

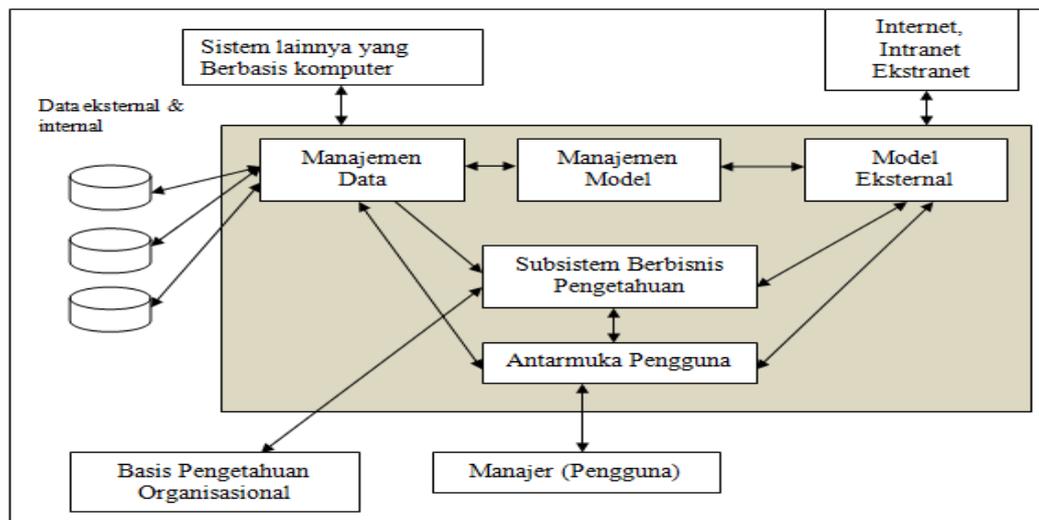
Pengguna berkomunikasi dan memerintahkan sistem pendukung keputusan melalui subsistem tersebut. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa

kontribusi unik dari sistem pendukung keputusan berasal dari interaksi yang intensif antara komputer dan pembuat keputusan. (Kusrini; 2007 : 25).

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 repositori pengetahuan perusahaan (bagian dari sistem manajemen pengetahuan), yang kadang-kadang disebut basis pengetahuan organisasional. (Kusrini; 2007 : 26).

Berdasarkan definisi, sistem pendukung keputusan harus mencakup tiga komponen utama dari DMBS, MBMS, dan antarmuka pengguna. Subsistem manajemen berbasis pengetahuan adalah opsional, tetapi bisa memberikan banyak manfaat karena memberikan intelegensi bagi ketiga komponen utama tersebut. Seperti pada semua sistem informasi manajemen, pengguna bisa dianggap sebagai komponen sistem pendukung keputusan. Komponen-komponen tersebut membentuk sistem aplikasi sistem pendukung keputusan yang bisa dikoneksikan ke intranet perusahaan, ekstranet, atau internet. Arsitektur dari sistem pendukung keputusan di tunjukkan dalam Gambar II.1. (Kusrini; 2007 : 26).



Gambar II.1. Arsitektur DSS
(Sumber : Kusri ; 2007 : 26)

II.2. Lokasi Event Promosi

II.2.1. Event

Menurut imam Hardiman (2006:39), *Event* adalah sebuah kegiatan yang bertujuan memperkenalkan atau mempopulerkan suatu hal. Sebuah *event* membutuhkan biaya yang besar karena meliputi : Sewa Lokasi, Honor artis pendukung, Biaya periklanan dan promosi, Biaya pembangunan set, dekor dan Biaya peorganisasian (*event organizer*), kepanitiaan sekretariat dan lain-lain. Sedangkan pengertian Event menurut Kamus Bahasa Inggris Indonesia adalah peristiwa atau kejadian. (Laporan Kerja Praktek Lapangan : Eka Widya Pratiwi : 2009).

II.2.2. Lokasi

Lokasi merupakan letak toko atau pengecer pada daerah yang strategis sehingga dapat memaksimalkan laba (Swastha, 2000). Pemilihan lokasi usaha yang tepat akan menentukan keberhasilan usaha tersebut di masa yang akan datang (Akhmad, 1996). Kolter (1994 : 96) mengartikan lokasi sebagai segala hal yang menunjukkan pada berbagai kegiatan yang dilakukan perusahaan untuk membuat produk tersedia dan dapat diperoleh bagi konsumen sasaran. Dari pendapat tersebut mengandung arti bahwa perusahaan hendaknya mengusahakan agar produk keluaran mereka tersedia dan terjangkau oleh populasi sasaran (konsumen). Lokasi berarti pula sebagai semua problem, fungsi dan lembaga yang berhubungan dengan usaha membawa produk yang tepat ke pasar target yang bersangkutan. (Naskah Publikasi Ilmiah : Langgeng Setyo Nugroho: 2015)

II.2.3. Promosi

Suatu perusahaan banyak aktivitas yang dilakukan tidak hanya menghasilkan produk atau jasa, menetapkan harga, dan menjual produk atau jasa, tetapi banyak aktivitas lainnya yang saling berkaitan satu dengan yang lainnya. Salah satunya adalah promosi, kegiatan promosi adalah salah satu bagian dari bauran pemasaran perusahaan, yang isinya memberikan informasi kepada masyarakat atau konsumen tentang produk atau jasa yang ditawarkan perusahaan. Tidak hanya itu, kegiatan promosi merupakan kegiatan komunikasi antara perusahaan dengan pelanggan atau konsumen.

Promosi dapat di definisikan sebagai suatu bentuk komunikasi pemasaran, komunikasi pemasaran adalah aktivitas pemasaran yang berusaha menyebarkan

informasi, mempengaruhi atau membujuk dan atau meningkatkan pasar sasaran atas perusahaan dan produknya agar bersedia menerima, membeli, dan loyal pada produk yang ditawarkan perusahaan yang bersangkutan. Tjiptono (2008 : 219). (Naskah Publikasi Ilmiah : Lenggeng Setyo Nugroho: 2015)

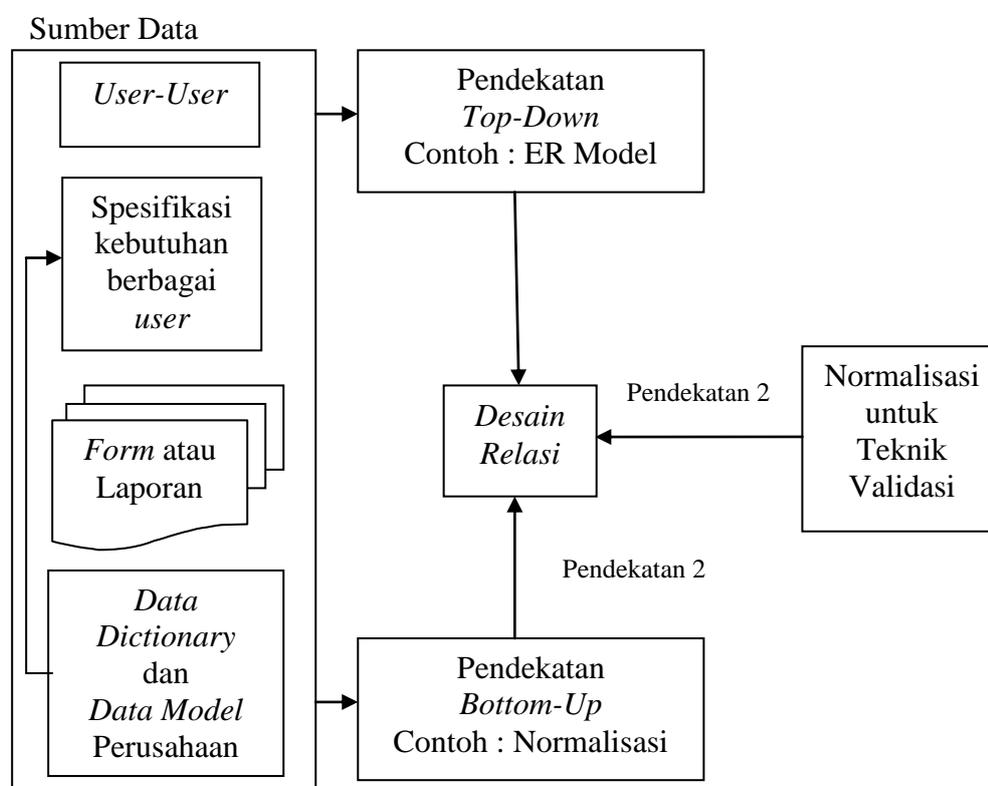
II.3. Normalisasi (*Normalization*)

Menurut Whitehorn (2003) normalisasi merupakan suatu proses yang memudahkan desain struktur tabel secara benar sehingga *query* yang tak dapat ditanyakan tidak muncul. Di samping itu, normalisasi cenderung meminimumkan duplikasi data di dalam suatu basis data. Ini memiliki keunggulan dalam mengurangi ruang simpan yang dibutuhkan maupun mempercepat *query* (Yanti Efendi, Jurnal Teknomatika, STMIK PalcomTech).

Menurut Indrajani (2015) normalisasi adalah suatu teknik dengan pendekatan *bottom-up* yang digunakan untuk membantu mengidentifikasi hubungan. Dimulai dari menguji hubungan, yaitu *functional dependencies* antara atribut. Tujuan normalisasi adalah mengidentifikasi kesesuaian hubungan yang mendukung data untuk memenuhi kebutuhan perusahaan. Adapun karakteristik hubungan tersebut mencakup:

1. Minimal jumlah atribut yang diperlukan untuk mendukung kebutuhan perusahaan.
2. Atribut dengan hubungan logika yang menjelaskan mengenai *functional dependencies*.
3. Minimal duplikasi untuk tiap atribut

Peranana normalisasi dalam perancangan basis data adalah dalam penggunaan pendekatan *bottom-up* dan teknik validasi. Teknik validasi digunakan untuk memeriksa, apakah struktur relasi yang dihasilkan oleh ER *modeling* itu baik atau tidak baik. Peranan normalisasi dalam perancangan basis data ditunjukkan pada Gambar II.2. berikut.



Gambar II.2. Peranan Normalisasi dalam Perancangan Basis Data
(Sumber : Indrajani ; 2015)

Di sini terlihat sumber data terdiri atas *user-user*, spesifikasi kebutuhan berbagai *user*, berbagai *form* atau laporan, *data dictionary*, dan *data model* perusahaan. Kemudian terdapat pendekatan *top-down* dan *bottom-up*, dimana pendekatan tersebut nantinya menghasilkan desain relasi. Lalu peranan normalisasi pada *bottom-up* dan teknik validasi.

Terdapat enam bentuk normal yang biasa digunakan, yaitu (Indrajani, 2015):

1. *First Norm Form*(1NF) atau Normalisasi Tingkat 1.
2. *Second Norm Form*(2NF) atau Normalisasi Tingkat 2.
3. *Third Norm Form*(3NF) atau Normalisasi Tingkat 3.
4. *Boyce-Cood Norm Form*(BCNF)
5. *Four Norm Form*(4NF)
6. *Five Norm Form*(5NF).

II.4. Microsoft Visual Studio 2010

Visual Basic diturunkan dari bahasa BASIC. Visual Basic terkenal sebagai bahasa pemrograman yang mudah digunakan terutama untuk membuat aplikasi yang berjalan di atas *platform Windows*. Pada tahun 90an, Visual Basic menjadi bahasa pemrograman yang paling populer dan menjadi pilihan utama untuk mengembangkan program berbasis Windows. Versi Visual Basic terakhir sebelum berjalan di atas .NET Framework adalah VB6 (Visual Studio 1998).

Visual Basic .NET dirilis pada bulan Februari tahun 2002 bersamaan dengan platform .NET Framework 1.0. Kini sudah ada beberapa versi dari Visual Basic yang berjalan pada platform .NET, yaitu VB 2002 (VB7), VB 2005(VB8), VB 2008 (VB9), dan yang terakhir adalah VB 2010 (VB10) yang dirilis bersamaan dengan Visual Studio 2010.

Selain Visual Basic 2010, Visual Studio 2010 juga mendukung beberapa bahasa lain, yaitu C#, C++, F# (bahasa baru untuk functional programming), IronPhyton, dan IronRuby (bahasa baru untuk dynamic programming). (Tugas Akhir Nurullah : 2012 : 20).

II.5. Microsoft SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang database. *SQL Server* adalah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. *Microsoft* merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut.

Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit (x86), yang biasanya digunakan untuk komputer dengan single prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
2. Versi 64-bit (x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core 2 Duo) dan sistem operasi 64

3. bit seperti Windows XP 64, Vista, dan Windows 7. (Wenny Widya, Iskandar Zulkarnaen, Jurnal Sistem Informasi, STMIK GI MDP : 3).

II.6. *Unified Modelling Language (UML)*

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan.

Berikut ini adalah beberapa diagram UML:

1. *Use Case Diagram*

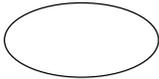
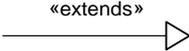
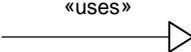
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
 - *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut ini adalah simbol-simbol yang ada pada *use case diagram* :

Tabel II.1. Simbol-simbol *Use Case Diagram*

Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor / <i>Actor</i></p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>Extends</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan</p>
<p>Generalisasi / <i>Generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan / <i>Include / Uses</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p>

(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 155-158)

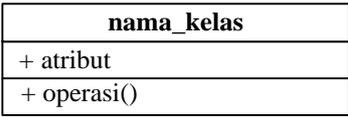
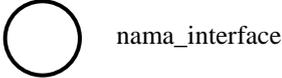
2. Class Diagram

Diagram kelas atau *Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut ini adalah simbol-simbol yang ada pada *class diagram* :

Tabel II.2. Simbol-Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi / <i>Generalization</i> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>)

(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 146-147)

3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan

- Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
- Rancangan menu yang ditampilkan pada perangkat lunak

Berikut ini adalah simbol-simbol yang ada pada *activity diagram* :

Tabel II.3. Simbol-simbol *Activity Diagram* (Diagram Aktivitas)

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>Decision</i> 	<i>Decision</i> , atau pilihan untuk mengambil keputusan.
Penggabungan / <i>Join</i> 	Arah tanda panah alur proses.
Status Akhir 	Titik akhir atau akhir dari aktivitas.

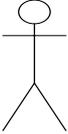
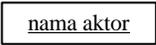
(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 162-163)

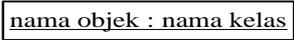
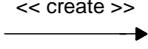
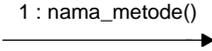
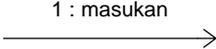
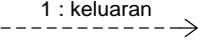
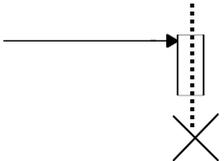
4. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Berikut adalah simbol-simbol yang ada pada diagram sekuen:

Tabel II.4. Simbol-simbol Diagram *Sequence*

Simbol	Keterangan
<p>Aktor / <i>Actor</i></p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>

<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya</p>
<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang di akhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 165-167)