

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

II.1.1. Sistem

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

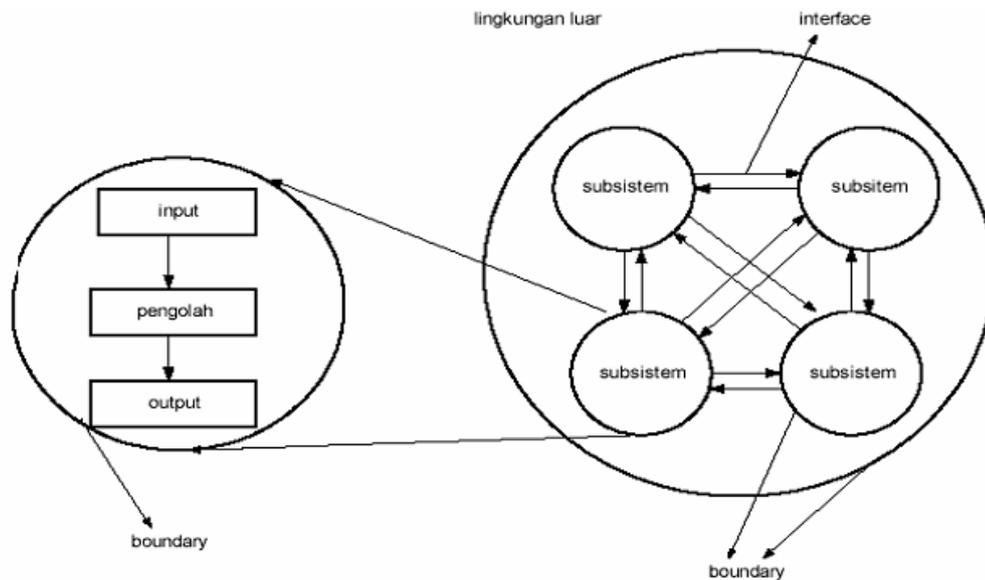
Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional (dengan tugas/fungsi khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses/pekerjaan tertentu. Sebagai contoh, sistem kendaraan terdiri dari: komponen starter, komponen pengapian, komponen penggerak, komponen pengerem, komponen kelistrikan- speedometer, lampu dan lain-lain. Komponen-komponen tersebut diatas memiliki tujuan yang sama yaitu untuk membuat kendaraan tersebut bisa dikendarai dengan nyaman dan aman. Contoh lain yaitu sistem perguruan tinggi, yang terdiri dari dosen, mahasiswa, kurikulum, dan lain-lain. Sistem ini bertujuan untuk menghasilkan mahasiswa-mahasiswa yang memiliki kemampuan di bidang ilmunya. (Kusrini; 2008: 4)

II.1.2. Konsep Dasar Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Menurut Jerry FithGerald ; sistem adalah suatu

jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. (Kusrini; 2008: 4)

Karakteristik sistem dapatlah digambarkan sebagai berikut :



Gambar II.1. Karakteristik Sistem
(Sumber : Kusrini; 2008: 5)

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Jadi, dapat dibayangkan jika dalam suatu sistem ada subsistem yang tidak berjalan/berfungsi sebagaimana mestinya. Tentunya

sistem tersebut tidak akan berjalan mulus atau mungkin juga sistem tersebut rusak sehingga dengan sendirinya tujuan sistem tersebut tidak tercapai.

2. Batas Sistem (*Boundary*)

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

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung (*Interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem lainnya dengan melalui penghubung. Dengan penghubung satu

subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

6. Keluaran (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada super sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

7. Pengolah (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi

menjadi laporan-laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

8. Sasaran (*Objectives*) atau Tujuan (*Goal*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Perbedaan suatu sasaran (*objectives*) dan suatu tujuan (*goal*) adalah, *goal* biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit. Bila merupakan suatu sistem utama, seperti misalnya sistem bisnis perusahaan, maka istilah *goal* lebih tepat diterapkan. Untuk sistem akuntansi atau sistem-sistem lainnya yang merupakan bagian atau subsistem dari sistem bisnis, maka istilah *objectives* yang lebih tepat. Jadi tergantung dari ruang lingkup mana memandang sistem tersebut. Seringkali tujuan (*goal*) dan sasaran (*objectives*) digunakan bergantian dan tidak dibedakan. (Kusrini; 2008: 8)

II.1.3. Informasi

Informasi adalah data yang dapat dianalogikan dengan data – data , yang belum di kelolah dan harus diolah untuk menjadi informasi yang akurat. Agar informasi yang penulis sajikan lebih bermanfaat maka terlebih dahulu dibuat aliran informasi yang lebih jelas dan lengkap. Berkaitannya dengan penyedia

informasi bagi manajemen dalam mengambil suatu keputusan, yang diperoleh harus berkualitas, maka kualitas dari informasi tergantung :

1. Akurat

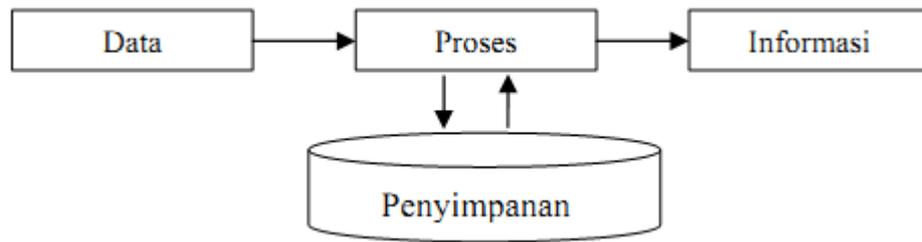
Akurat berarti bahwa informasi harus bebas dari kesalahan - kesalahan dan tidak biasa (menyesatkan) dan jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerimaan informasi kemungkinan banyak terjadi gangguan yang dapat merubah informasi atau merusak informasi tersebut.

2. Relevansi

Relevansi berarti bahwa informasi benar – benar berguna bagi suatu tindakan dan keputusan oleh seseorang

3. Tepat waktu

Tepat waktu berarti bahwa informasi yang datang pada penerimaan tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan dalam pengambilan keputusan. Dewasa ini mahalnya nilai informasi disebabkan harus cepatnya informasi itu di dapat, Untuk lebih jelasnya informasi merupakan hasil atau output dari proses informasi data (Kusrini : 2008: 15) . Hal ini dapat dilihat seperti gambar dibawah ini :



Gambar II.2. Proses Data Menjadi Informasi
(Sumber : Kusriani; 2008: 15)

II.2. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tidak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Kusriani; 2007: 15-16).

DSS (*Decision Support System*) atau Sistem Pendukung Keputusan Sistem Pendukung Keputusan biasanya dibangun untuk mendukung solusi suatu masalah atau untuk mengevaluasi suatu peluang, hal tersebut disebut dengan aplikasi DSS. Aplikasi DSS digunakan dalam pengambilan keputusan.

Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah dan dapat menggabungkan pemikiran pengambil keputusan. DSS lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan criteria yang kurang jelas (Kurini; 2007: 16).

Secara umum, sistem pendukung keputusan mencakup beberapa komponen utama yaitu DBMS, MBMS dan antarmuka pengguna. Adapun juga komponen opsional yang disebut subsistem manajemen berbasis pengetahuan. Komponen-komponen tersebut dapat dijelaskan sebagai berikut:

1. Manajemen Data (DBMS)

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

2. Manajemen Model (MBMS)

Merupakan paket perangkat lunak yang memasukkan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lain yang memberika 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).

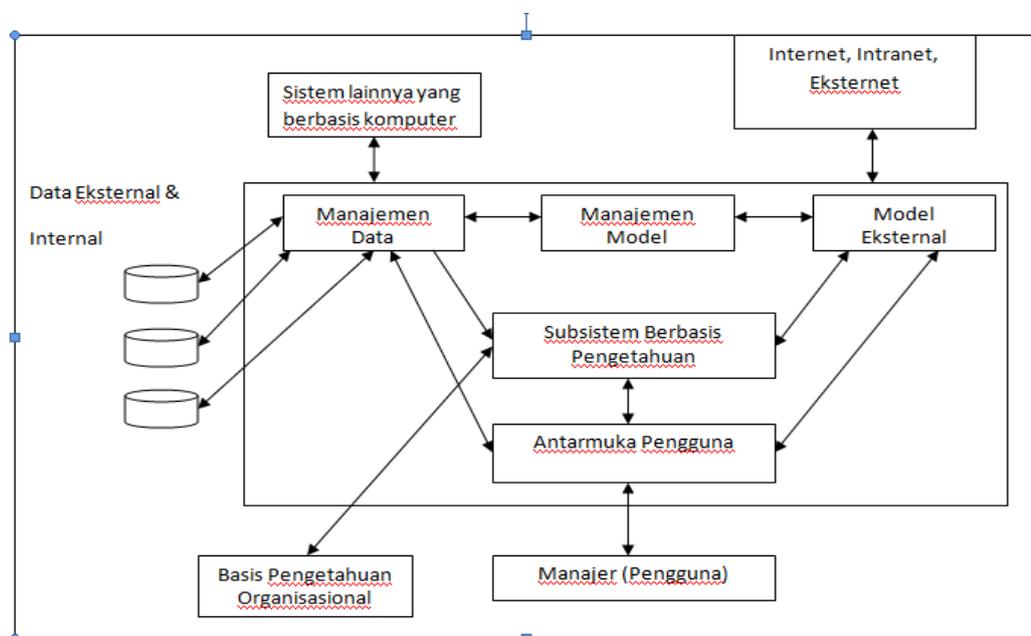
3. Antarmuka Pengguna

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 computer dan pembuat keputusan (Kusrini; 2007: 25).

4. 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 (Kusrini; 2007: 26).



Gambar II.3. Arsitektur DSS
(Sumber : Kusrini; 2007: 26)

II.3. Simple Additive Weighting

Metode *Simple Additive Weighting* (SAW) merupakan yang juga dikenal dengan metode penjumlahan berbobot, konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Rudi Hartoyo, *Jurnal ISSN*, No. 3, Agustus 2013: volume IV).

Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi Multiple Attribute Decision Making (MADM). MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu.

Ada beberapa tahapan untuk menyelesaikan suatu kasus menggunakan metode SAW yaitu :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i .
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria(C_i), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga

diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi.

Berikut formula rumus yang akan digunakan pada metode Simple Additive Weighting :

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Gambar II.4. Formula Rumus SAW

(Sumber : Rudi Hartoyo, Jurnal ISSN, No. 3, Agustus 2013: volume IV)

Dimana :

r_{ij} = rating kinerja ternormalisasi

Max_{ij} = nilai maksimum dari setiap baris dan kolom

Min_{ij} = nilai minimum dari setiap baris dan kolom

X_{ij} = baris dan kolom dari matriks

Dengan r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$.

Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai berikut :

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Gambar II.5. Nilai Preferensi SAW

(Sumber : Rudi Hartoyo, Jurnal ISSN, No. 3, Agustus 2013: volume IV)

Dimana :

V_i = Nilai akhir dari alternatif

w_j = Bobot yang telah ditentukan

r_{ij} = Normalisasi matriks

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

II.4. 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 system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini:

1. Versi Compact, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi desktop pada SQL Server 2000. Versi ini juga digunakan pada handled drvice seperti Pocket PC, PDA, SmartPhone, Tablet PC.
2. Versi Express, ini adalah versi “Ringan” dari semua versi yang ada tetapi versi ini berbeda dengan versi compact) dan paling cocok untuk latihan para pengembang aplikasi. Versi ini memuat Hal – 4 Express Manager standar, integrasi dengan CLR dan XML. (Ananda; Wendy, dkk: 2014: 3)

II.5. UML (*Unified Modelling Language*)

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles, 2003: bab 1) mengatakan sebagai bahsa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model – model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya ? Bagaimana sistem mengatasi error yang terjadi ? Bagaiman

kemanan terhadap sistem yang kita buat ? Dan sebagainya dapat dijawab dengan UML. (Prabowo Pudjo Widodo & Herlawati : 2011: 6)

UML dapat diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

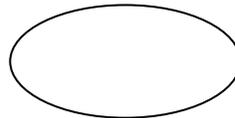
1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis.(Prabowo Pudjo Widodo & Herlawati; 2011: 10)

1. Use Case Diagram

Menurut (Pilone, 2005 : bab 7.1) use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan (Whitten, 2004: 258) mengartikan use case sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas

bisnis tunggal. Use case digambarkan dalam bentuk ellips/oval. (Prabowo Pudjo Widodo & Herlawati; 2011: 21-22)



Gambar II.6. Simbol Use Case
(Sumber : Prabowo Pudjo Widodo & Herlawati; 2011: 22)

Elemen use case terdiri dari :

- a. Diagram use case, disertai dengan narasi dan skenario.
- b. Aktor (actor), mendefinisikan entitas diluar sistem yang memakai sistem.
- c. Asosiasi (associations), mengindikasikan aktor mana yang berinteraksi dengan use case dalam suatu sistem.
- d. << include >> dan << extend >>. Merupakan indikator yang menggambarkan jenis relasi dan interaksi antar use-case.
- e. Generalisasi (generalization), menggambarkan hubungan turunan antar use case atau antar aktor. (Prabowo Pudjo Widodo & Herlawati; 2011: 35)

2. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik forward engineering maupun reverse engineering memanfaatkan diagram ini. Forward engineering adalah proses perubahan model menjadi kode program

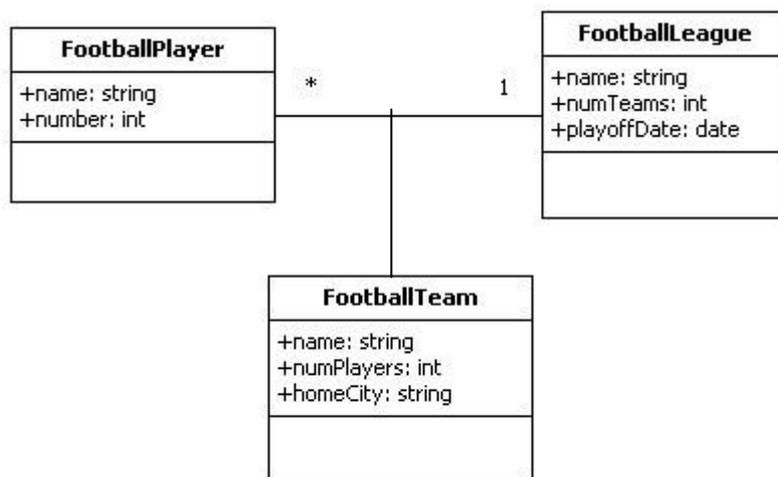
sedangkan reverse engineering sebaliknya merubah kode program menjadi model. (Prabowo Pudjo Widodo & Herlawati; 2011: 37).

Diagram kelas merupakan kumpulan kelas-kelas objek. Oleh karena itu pengertian kelas sangat penting sebelum merancang diagram kelas. (Whitten, 2004: 410) mengartikan kelas sebagai satu set objek yang memiliki atribut kelas objek (objek class). Secara alami, objek yang berupa buku analisis desain dan buku pemrograman terstruktur kita kelompokkan dalam satu kelas, yaitu kelas buku. Kedua objek memiliki atribut dan perilaku yang serupa. Contohnya, kedua objek mungkin memiliki atribut yang serupa seperti nomor ISBN, Judul, tanggal penerbitan, edisi, dan sebagainya. Demikian juga, kedua objek memiliki perilaku yang sama misalnya membuka dan menutup. (Prabowo Pudjo Widodo & Herlawati; 2011: 39).

Kelas menggambarkan suatu group yang memiliki kesamaan dan perilaku. Kelas merupakan cetak biru suatu objek dalam suatu sistem orientasi objek. Dapat dikatakan kelas adalah sejenis alat pengklasifikasi. Sebagai contoh volkswag, toyota dan ford merupakan kumpulan mobil sehingga kita dapat mengelompokkannya dalam menyatakan konsep yang dapat dilihat maupun abstrak.

Kelas dinyatakan dalam kotak yang terbagi menjadi beberapa kompartemen. Kompartemen adalah area dalam kelas yang berisi informasi. Kompartemen pertama bernama kelas, berikutnya atribut dan yang terakhir operasinya. Kompartemen-kompartemen itu bisa

disembunyikan untuk memperjelas diagram yang kita buat. (Prabowo Pudjo Widodo & Herlawati; 2011: 40)

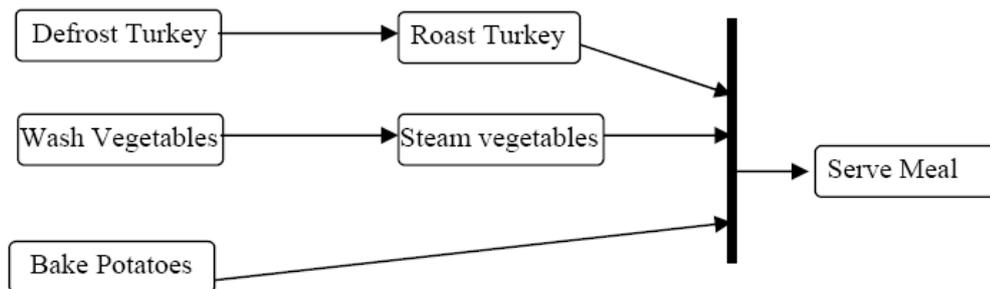


Gambar II.7. Contoh Class Diagram
(Sumber : Prabowo Pudjo Widodo & Herlawati; 2011: 69)

3. Activity diagram

Diagram aktifitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan model bisnis juga. Diagram aktifitas menunjukkan aktifitas sistem dalam bentuk aksi-aksi. Ketika digunakan dalam pemodelan software, diagram aktifitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya call. Sedangkan bila dimodelkan dalam pemodelan bisnis, diagram ini menggambarkan aktifitas yang dipicu oleh kejadian-kejadian di luar seperti pemesanan atau kejadian-kejadian internal. (Prabowo Pudjo Widodo & Herlawati; 2011: 143)

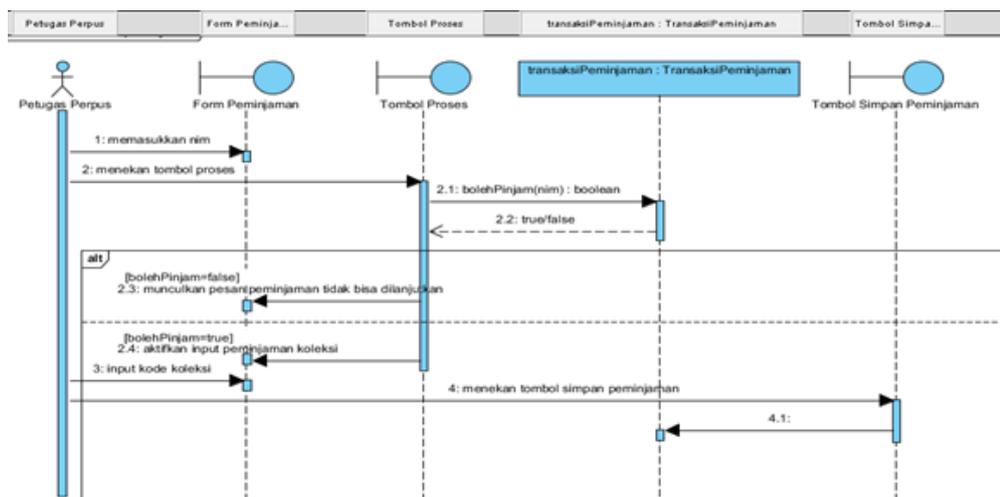
Prepare Meal



Gambar II.8. Contoh Activity Diagram Sederhana
 (Sumber : Prabowo Pudjo Widodo & Herlawati; 2011: 163)

4. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sebuah contoh objek dan pesan yang diletakkan diantara objek-objek ini didalam *use case*. Diagram sequence dimaksudkan untuk mengembangkan komunikasi antara objek, bukan memanipulasi data saat berkomunikasi. (Prabowo Pudjo Widodo & Herlawati; 2011: 174)



Gambar II.9. Contoh Sequence Diagram
(Sumber : Haviluddin, Jurnal Informatika Mulawarman, No. 1, Februari 2011: volume VI)

II.6. Visual Basic 2010

II.6.1. Pengertian Visual Basic

Yang dimaksud dengan visual basic adalah suatu produk software terbaru yang dikeluarkan oleh Microsoft Cooperation, yaitu microsoft Visual Studio 2010 dan mempunyai beberapa fitur terbaru dan mempunyai suatu bentuk yang menawarkan IDE (Integrated Development Environment) dengan menggunakan model COM. Visual Basic juga merupakan turunan dari bahasa pemrograman BASIC sehingga mudah untuk membantu dalam mengembangkan program dan aplikasi.

Penemu visual basic adalah Alan Cooper, yaitu pelopor GUI dan Bapak Visual Basic pada tahun 1988. Kegunaan visual basic adalah :

1. Untuk membuat program berbasis windows.

2. Membuat objek-objek pembantu program seperti kontrol ActiveX, file help, dan lain lain.
3. Menguji program (debugging) dan menghasilkan program berekstensi EXE bersifat executable/dapat langsung dijalankan.
4. Kemudahan desain form. Visual basic unggul dengan kemudahan desain form untuk menggunakan komponen activeX yang dibuat oleh pihak lain.

II.6.2. Sejarah 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 sensitif.

Sejarah *Visual Basic* berawal dari perkembangan bahasa BASIC di Amerika Serikat pada awal tahun 1960-an. Pada tahun 1982 IBM/PC diperkenalkan pada masyarakat yang didalamnya, disertakan pula bahasa BASIC yang dikenal juga *QuickBasic / QBASIC*. Pada tahun 1990-an DOS digantikan oleh Windows. Microsoft akhirnya membuat BASIC versi Windows yang dikenal sebagai *Microsoft Visual Basic*. Awal perkembangan *Visual Basic* yaitu :

- Dirintis proyek “Thunder”
- Pada tahun 1991, *Visual Basic 1.0* dirilis untuk windows pada Comdex.
- Pada November 1992, *Visual Basic 2.0* dirilis dengan pemrograman yang cukup mudah digunakan serta kecepatannya telah dimodifikasi.
- Pada tahun 1993, *Visual Basic 3.0* yang dibagi menjadi versi standard dan professional.
- Pada tahun 1995, merupakan versi pertama yang dapat membuat windows 32 bit sebaik versi 16 bit nya yaitu *Visual Basic 4.0*.
- Pada tahun 1997, *Visual Basic 5.0* yang dirilis microsoft merilis secara eksklusif untuk versi windows 32 bit.
- Pertengahan 1998, *Visual Basic 6.0* dengan perbaikan beberapa cakupan termasuk kemampuan dalam pembuatan Aplikasi *Web-based*.
- Pada tahun 2002, dirilis *Visual Basic.Net (VB 7)* dengan bahasa yang sangat powerful namun bahasanya sangat berbeda dari yang sebelumnya.
- Pada Tahun 2003, dirilis *Visual Basic.Net 2003 (VB 7.1)* dengan menggunakan *Net framework* versi 1.1.
- Pada Tahun 2005 (VB 8.0), *Visual Basic 2005* yang merupakan lanjutan dari *Visual Basic .NET*.
- *Edit & Continue* , pada *Visual Basic 2005* kita boleh melakukan perubahan kode.
- *Visual Basic “Orcas” (VB 9.0)* , pada tahun 2007 dengan penambahan fitur – fitur.

- *Visual Basic 'VBx' (VB 10.0)*, menawarkan dukungan untuk *Dynamic Language Runtime*. (“Sejarah Perkembangan Visual Basic”, <http://ademaesyaputra.wordpress.com/author/ademaesyaputra/page/2/>, http://id.wikipedia.org/wiki/Visual_Basic, <http://aerotechno2.blog.com/kelelahan-visual-basic/>, <http://ninysnurindahsari.blogspot.com/2011/09/fungsi-dan-kegunaan-visual-basic.html> akses 25 Mei 2014).