BAB II

TINJAUAN PUSTAKA

II.1. Teori Sistem

Menurut Kusrini (2010:5), Kata Sistem mempunyai beberapa pengertian, tergantung dari sudut mana kata tersebut didefenisikan. 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 didefenisikan 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 didefenisikan 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 mendefisikan 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 berbentuk suatu kesatuan hingga tujuan sistem dapat tercapai.

II.1.1. Karakteristik sistem

Menurut Kusrini (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 berupaya 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 (*Environtment*)

   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.2. Sistem Pakar

II.2.1. Pengertian Sistem Pakar

Sistem Pakar (*Expert System*) adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pernyataan dan memecahkan suatu masalah. (T. Sutojo; 2011: 13)

Istilah sistem pakar berasal dari istilah *knowledge – based expert system*. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan kedalam komputer. Seorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant*. Berikut adalah beberapa pengertian sistem pakar (T.Sutojo,dkk ; 2011 : 160).

1. Turban

"Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah - masalah yang biasanya membutuhkan kepakaran atau keahlian manusia".

2. Jackson

"Sistem pakar adalah program komputer yang merepresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan kepakaran atau keahlian manusia".

3. Lugar dan Stubblefield

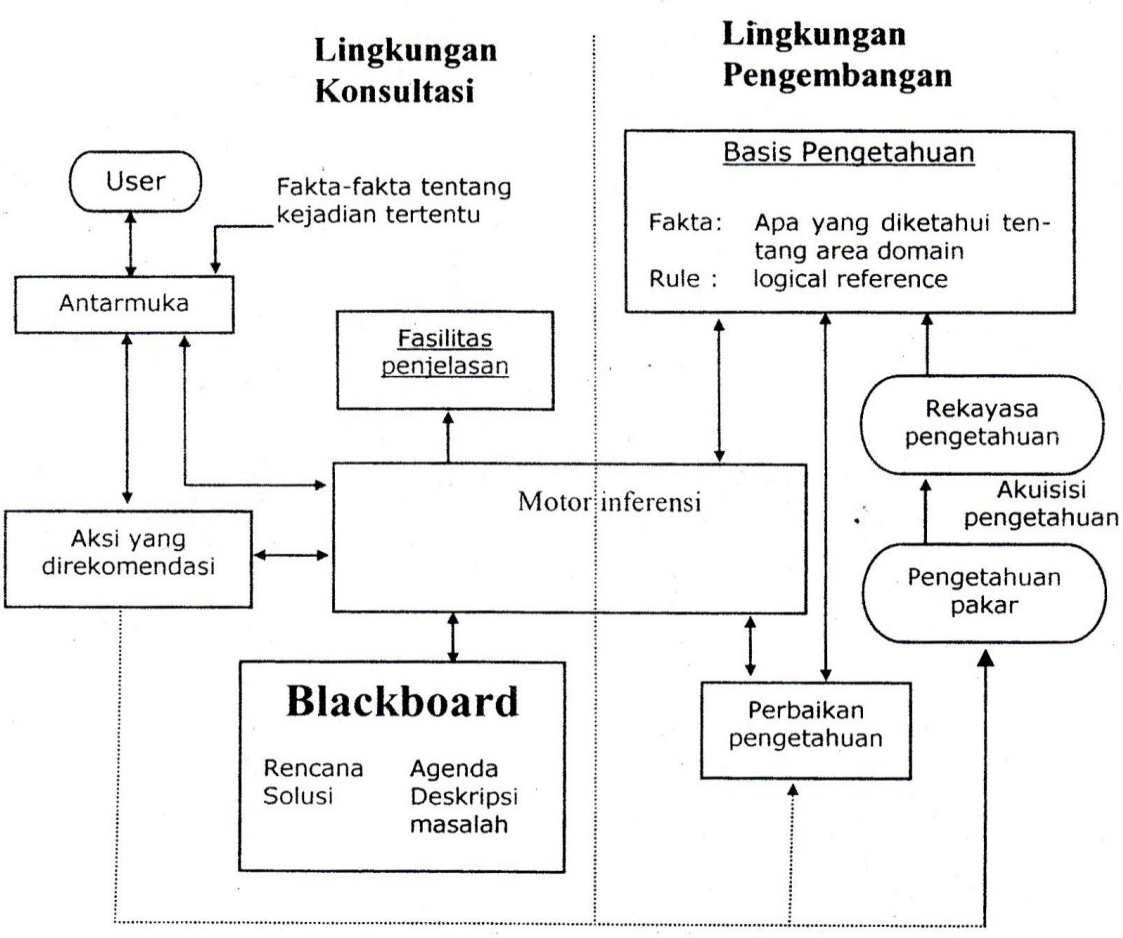
"Sistem pakar adalah program yang berbasiskan pengetahuan yang menyediakan solusi 'kualitas pakar' kepada masalah – masalah dalam bidang (domain) yang spesifik".

Jadi, sistem pakar adalah sistem informasi yang berisi dengan pengetahuan dari pakar sehingga dapat digunakan sebagai media konsultasi untuk memecahkan masalah

II.2.3. Struktur Sistem Pakar

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Lingkungan

konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari Sistem Pakar layaknya berkonsultasi dengan seorang pakar. Gambar II.1 menunjukkan komponen-komponen yang penting dalam sebuah sistem pakar.



Gambar II.1 : Struktur Sistem Pakar

Sumber: (T. Sutojo; 2011: 167)

Keterangan:

1. Akuisisi Pengetahuan

   Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan manaruhnya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan). Sumber-sumber pengetahuan bisa diperoleh dari pakar, buku, dokumen, multimedia, basis data, laporan riset khusus, dan informasi yang terdapat di web.

2. Basis Pengetahuan (*Knowledge Base*)

   Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu:

   a. Fakta, misalnya situasi, kondisi, atau permasalahan yang ada.
   b. Rule (aturan), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

3. Mesin Inferensi (*Inference Engine*)

   Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan proses

penalaran. Ada tiga teknik pengendalian yang digunakan, yaitu *forward chaining, backward chaining,* dan gabungan dari kedua teknik tersebut.

4. Daerah Kerja (*Blackboard*)

   Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi, sistem pakar membutuhkan *Blackboard* yaitu area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *Blackboard,* yaitu:

   a. Rencana : bagaimana menghadapi masalah.

   b. Agenda : aksi-aksi potensial yang sedang menunggu untuk dieksekusi.

   c. Solusi : calon aksi yang akan dibangkitkan.

5. Antarmuka Pengguna (*User Interface*)

   Digunakan sebagai media komunikasi antara pengguna dan sistem pakar, komunikasi ini paling bagus bila disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu, dan formulir elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

6. Subsistem Penjelasan (*Explanation Subsystem/Justifire*)

   Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

7. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)

   Kemampuan memperbaiki pengetahuan (*knowledge refining system*) dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari

kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang. Kemampuan evaluasi diri seperti itu diperlukan oleh program agar dapat menganalisa alasan-alasan kesuksesan dan kegagalannya dalam mengambil kesimpulan. Dengan cara ini basis pengetahuan yang lebih baik dan penalaran yang lebih efektif akan dihasilkan.

8. Pengguna (*User)*

   Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) daru berbagai permasalahan yang ada.

II.3. Penyakit Gagal Ginjal

Penyakit Gagal Ginjal adalah suatu penyakit dimana fungsi organ ginjal mengalami penurunan hingga akhirnya tidak lagi mampu bekerja sama sekali dalam hal penyaringan pembuangan elektrolit tubuh, menjaga keseimbangan cairan dan zat kimia tubuh seperti *sodium* dan *kalium* di dalam darah atau produksi *urin*.

Penyakit gagal ginjal ini dapat menyerang siapa saja yang menderita penyakit serius atau terluka dimana hal itu berdampak langsung pada ginjal itu sendiri. Penyakit gagal ginjal lebih sering dialami mereka yang berusia dewasa, terlebih pada kaum lanjut usia.

Gagal ginjal dibagi menjadi dua bagian besar yakni gagal ginjal akut (*acute renal failure = ARF*) dan gagal ginjal kronik (*chronic renal failure =*

CRF). Pada gagal ginjal akut terjadi penurunan fungsi ginjal secara tiba-tiba dalam waktu beberapa hari atau beberapa minggu dan ditandai dengan hasil pemeriksaan fungsi ginjal (*ureum* dan *kreatinin* darah) dan kadar *urea* nitrogen dalam darah yang meningkat. Sedangkan pada gagal ginjal kronis, penurunan fungsi ginjal terjadi secara perlahan-lahan. Sehingga biasanya diketahui setelah jatuh dalam kondisi parah. Gagal ginjal kronik tidak dapat disembuhkan. Pada penderita gagal ginjal kronik, kemungkinan terjadinya kematian sebesar 85 %.

II.4. *Dempster-Shafer*

Metode *Dempster-Shafer* pertama kali diperkenalkan oleh Dempster, yang melakukan percobaan model ketidakpastian dengan *range* probabilities dari pada sebagai probabilitas tunggal. Kemudian pada tahun 1976 Shafer mempublikasikan teori Dempster itu pada sebuah buku yang berjudul *Mathematical Theory Of Evident*. *Dempster-Shafer Theory Of Evidence*, menunjukkan suatu cara untuk memberikan bobot kenyakinan sesuai fakta yang dikumpulkan. Pada teori ini dapat membedakan ketidakpastian dan ketidaktahuan. Teori *Dempster-Shafer* adalah representasi, kombinasi dan propogasi ketidakpastian, dimana teori ini memiliki beberapa karakteristik yang secara instutitif sesuai dengan cara berfikir seorang pakar, namun dasar matematika yang kuat.

Ada berbagai macam penalaran dengan model yang lengkap dan sangat konsisten, tetapi pada kenyataannya banyak permasalahan yang tidak dapat terselesaikan secara lengkap dan konsisten. Ketidak konsistenan yang tersebut adalah akibat adanya penambahan fakta baru. Penalaran yang seperti itu disebut

dengan penalaran *non monotonis*. Untuk mengatasi ketidak konsistenan tersebut maka dapat menggunakan penalaran dengan teori *Dempster-Shafer*. Secara umum teori Dempster-Shafer ditulis dalam suatu interval.

Penulisan umum: [*belief, plausibility*]

1. *Belief* (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian.
2. *Plausibility* (P1) dinotasikan sebagai:

   Pl(s) = 1 – Bel(¬s)

   Plausibility juga bernilai 0 sampai 1. Jika yakin akan ¬s, maka dapat dikatakan bahwa Bel (¬s) = 1, dan PI (¬ s) = 0.

Pada teori *Dempster-Shafer* dikenal adanya *frame of discernment* yang dinotasikan dengan θ. Frame ini merupakan semesta pembicaraan dari sekumpulan hipotesis. Tujuannya adalah mengkaitkan ukuran kepercayaan elemen-elemen θ. Tidak semua *evidence* secara langsung mendukung tiap-tiap elemen. Untuk itu perlu adanya probabilitas fungsi densitas (m). Nilai m tidak hanya mendefinisikan elemen-elemen θ saja, namun juga semua subsetnya. Sehingga jika θ berisi n elemen, maka subset θ adalah 2n. Jumlah semua m dalam subset θ sama dengan 1. Apabila tidak ada informasi apapun untuk memilih hipotesis, maka nilai: m{θ} = 1,0.

Apabila diketahui X adalah subset dari θ, dengan m1 sebagai fungsi densitasnya, dan Y juga merupakan subset dari θ dengan m2 sebagai fungsi

densitasnya, maka dapat dibentuk fungsi kombinasi m1 dan m2 sebagai m3, yaitu:

$$M_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X).m_2(Y)}{1 - \kappa} \quad \text{..... (1)}$$

Keterangan:

m1 (X) = *mass function dari evidence* X

m2 (Y) = *mass function dari evidence* Y

m3 (Z) = *mass function dari evidence* Z

κ = jumlah *conflict evidence* (Dewi Ermayani; 2012: 3)

II.4.1 Mesin Inferensi (*Inference Engine*)

Ada 2 cara yang dapat dikerjakan dalam melakukan inferensi, yaitu:

1. *Forward Chaining* (Pelacakan ke depan)

   Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji hipotesis.

2. *Backward Chaining* (Pelacakan ke belakang)

   Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN dulu). Dengan kata lain, penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut, harus dicari fakta-fakta yang ada dalam basis pengetahuan (Yasidah Nur Istiqomah ; 2013 : 35).

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

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standar. Sebagai bahasa, 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 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? Bagaimana keamanan terhadap sistem yang kita buat? Dan sebagainya dapat dijawab dengan UML. (Prabowo Pudjo Widodo; 2011: 6)

UML diaplikasikan untuk maksud tertentu, biasanya antara lain:

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. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya. (Prabowo Pudjo Widodo; 2011: 6)

II.5.1. Jenis-Jenis Diagram UML

Berikut ini adalah mengenai berbagai diagram UML serta tujuan dan simbol-simbolnya, yaitu:

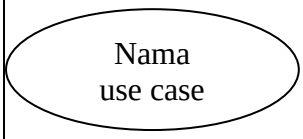
1. Diagram *Use Case (Use Case Diagram)*

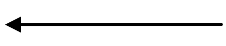
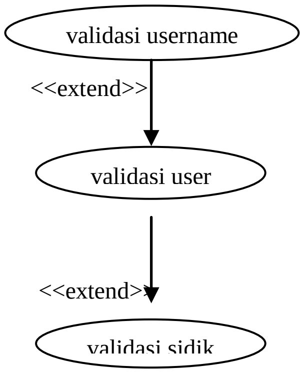
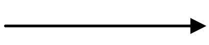
   Diagram *Use case* merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan

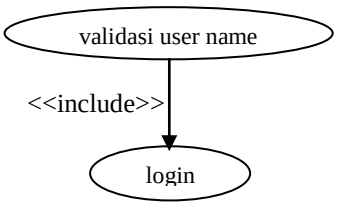
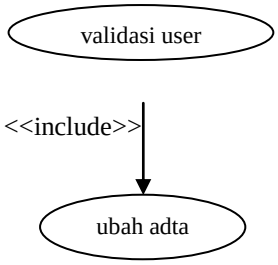
sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

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

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

Simbol	Deskripsi
Use case  	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dengan menggunakan kata kerja di awal frase nama *use case*.
Aktor/*actor*  Nama actor	Orang, proses, atau lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu adalah orang, biaanya dinyatakan menggunakan kata benda di awal fase nama aktor.
Asosiasi/*association* ———	Komunikasi antara aktor dan use case yang berpartisipasi pada *use case* atau *use case* miliki interaksi dengan actor

Ekstensi/*extend*  <<extend>>	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang di tambahkan, misal  Arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.
Include <<include>> 	Relasi *use case* tambahan sebuah *use case* di mana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. Ada dua sudut pandang yang cukup besar mengenai *include* di *use case*: • *include* berarti *use case* yang ditambahkan akan

	selalu di panggil saat use case tambahan dijalankan. Contoh:  • Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan, missal pada kasus berikut:  Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

Sumber: (Rosa A.S; 2013: 156)

2. Diagram Kelas (*Class Diagram*)

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

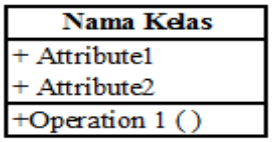
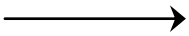
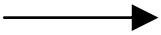
- Atribut merupakan variable-variabel yang dimiliki suatu kelas.

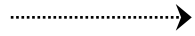
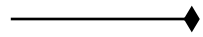
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. (Rosa A.S; 2013: 143)

Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi, perancangan, dan perangkat lunak sinkron. (Rosa A.S; 2013: 144)

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/*interface* 	Sama dengan konsep *interface* dalam pemograman berorientasi objek
Asosiasi/*association* 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah/*directed asosiasi* 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)

Kebergantungan/*dependency* 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi/aggregiation 	Relasi antarkelas dengan makna semua-bagian (*whole-part*)

Sumber: (Rosa A.S; 2013: 146)

3. Diagram Aktivitas (*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. (Rosa A.S; 2013: 161)

4. Diagram Rangkaian (*Sequence Diagram*)

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek. Oleh karena itu untuk menggambar 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*. (Rosa A.S; 2013: 165)

II.6. *Entity Relationship Diagram* (ERD)

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. (Rosa A.S; 2013: 50)

Entity Relationship Diagram (ERD) adalah menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika.

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi**.**ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu :

1. Entiti, merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.
2. Atribut, Setiap entitas pasti mempunyai elemen yang disebut *atribut* yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar *atribut* diwakili oleh simbol elips.
3. Hubungan / Relasi, Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut: Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

a) ***Satu ke satu (One to one)***, Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

b) ***Satu ke banyak (One to many),*** Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

c) ***Banyak ke banyak (Many to many),*** Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

II.7. Visual Basic 2010

Visual Basic 2010 merupakan salah satu bagian dari produk pemograman terbaru yang dikeluarkan oleh Microsoft, yaitu Microsoft Visual Studio 2010. Visual Studio merupakan produk pemrograman andalan dari Microsoft Corporation, dimana di dalamnya berisi beberapa jenis IDE pemrograman seperti Visual Basic, Visual C++, Visual Web Developer, Visual C#, dan Visual F#. Semua IDE pemrograman tersebut sudah mendukung penuh implementasi .Net Framework terbaru, yaitu .Net Framework 4.0 yang merupakan pengembangan dari .Net Framework 3.5. (Wahana Komputer; 2011: 2)

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu Visual Basic 2008. Bahasa Visual Basic sendiri awalnya berasal dari bahasa pemrograman yang sangat popular di kalangan

programmer computer, yaitu bahasa BASIC, yang oleh Microsoft diadaptasi dalam program Microsoft Quick BASIC. Seiring dengan berkembangnya teknologi komputasi dan desain, Microsoft mengeluarkan produk yang dinamakan Microsoft Visual Studio dengan Visual Basic di dalamnya. Saat ini versi Microsoft Visual Studio yang beredar adalah versi 10 yang popular dengan nama Microsoft Visual Studio 2010, yang di dalamnya termasuk Microsoft Visual Basic 2010. (Wahana Komputer; 2011: 3)

II.8. SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang database. SQL Server adalah sebuah 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. (Wahana Komputer; 2010: 2)

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan computer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. (Wahana Komputer; 2010: 24)