BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. Beberapa kelebihan sistem pakar adalah meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia, membuat seorang yang awam bekerja seperti layaknya seorang pakar, mampu menangkap pengetahuan dan kepakaran seseorang. (Jurnal Teknik Informatika ; Purnama Ramadhani, dkk ; 2012 : 1).

Sistem pakar (*Expert System*)merupakan solusi AI bagi masalah pemrograman pintar (*Intelligent*).Profesor Edward Feigenbaum dari *Stanford University* yang merupakan *pionir* dalam teknologi sistem pakar mendefenisikan sistem pakar sebagai sebuah program komputer pintar (*intelligentcomputer program*) yang memanfaatkan pengetahuan (*knowledge*) dan prosedur inferensi (*inference procedure*) untuk memecahkan masalah yang cukup sulit hingga membutuhkan keahlian khusus dari manusia.

Dengan kata lain, sistem pakar adalah sistem komputer yang ditujukan untuk meeniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah.

Pakar atau ahli (*expert*) didefenisikan sebagai seseorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang.Seorang pakar dapat memecahkan masalah yang tidak mampu dipecahkan kebanyakan orang. Dengan kata lain, dapat memecahkan suatu masalah dengan lebih efisien namun bukan berarti lebih murah. Pengetahuan yang dimuat ke dalam sistem pakar dapat berasal dari seorang pakar atau pun pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang dipublikasikan lainnya, serta orang yang memiliki pengetahuan meskipun bukan ahli. Istilah sistem pakar (*expert system*). Sering disinonimkan dengan sistem berbasis pengetahuan (*knowledge-based system*) atau sistem pakar berbasis pengetahuan (*knowledge based expert system*). (Rosnelly ; 2012 : 2).

II.1.1. Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya, seperti :

a. Meningkatkan ketersediaan (*increased availability*). Kepakaran atau keahlian menjadi tersedia dalam sistem komputer. Dapat dikatakan bahwa sistem pakar merupakan produksi kepakaran secara masal (*massproduction*).
b. Mengurangi biaya (*reduced cost*). Biaya yang diperlukan untuk menyediakan keahlian per satu orang *user* menjadi berkurang.
c. Mengurangi bahaya (*reduced danger*). Sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.

d. Permanen (*permanence*). Sistem pakar dan pengetahuan yang terdapat di dalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan, dan pengetahuannya hilang saat sang pakar meninggal dunia.
e. Keahlian multiple (*multiple expertise*). Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinyu menyelesaikan suatu masalah setiap saat. Tingkat keahlian atau pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar.
f. Meningkatkan kehandalan (*increased reliability*). Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku jika sistem dibuat oleh salah seorang pakar, sehingga akan selalu sama dengan pendapat pakar tersebut kecuali jika sang pakar melakukan kesalahan yang mungkin terjadi pada saat tertekan atau stres.
g. Penjelasan (*explanation*). Sistem pakar dapat menjelaskan detail proses penalaran (*reasoning*) yang dilakukan hingga mencapai suatu kesimpulan. Seorang pakar mungkin saja terlalu lelah, tidak bersedia atau tidak mampu melakukannya setiap waktu. Hal ini akan meningkatkan tingkat kepercayaan bahwa kesimpulan yang dihasilkan adalah benar.

h. Respon yang cepat (*fast response*). Respon yang cepat atau *real time* diperlukan pada beberapa aplikasi. Meskipun bergantung pada *hardware* dan *software* yang digunakan, namun sistem pakar relative memberikan respon yang lebih cepat dibandingkan seorang pakar.

i. Stabil, tidak emosional, dan memberikan respon yang lengkap setiap saat (*steady, unemotional, and complete response at all times*). Karakteristik ini diperlukan pada situasi *real-time* dan keadaan darurat (*emergency*) ketika seorang pakar mungkin tidak berada pada kondisi puncak disebabkan oleh stress atau kelelahan.

j. Pembimbing pintar (*intelligent tutor*). Sistem pakar dapat berperan sebagai *intelligent tutor* dengan memberikan kesempatan pada *user* untuk menjalankan contoh program dan menjelaskan proses *reasoning* yang dilakukan.

k. Basis data cerdas (*intelligent database*).Sistem pakar dapat digunakan untuk mengaksesbasis data secara cerdas.(Rosnelly ; 2012 : 5).

II.1.2. Konsep Umum Sistem Pakar

Pengetahuan yang dimiliki sistem pakar direpresentasikan dalam beberapa cara. Salah satu metode yang paling umum digunakan adalah tipe *rule* menggunakan format IF THEN. Banyak sistem pakar yang dibangun dengan mengekspresikan pengetahuan dalam bentuk *rules*. Bahkan, pendekatan berbasis

pengetahuan (*knowledgebased approach*) untuk membangun sistem pakar telah mematahkan pendekatan awal yang digunakan pada sekitar tahun 1950-an dan 1960-an yang menggunakan teknik penalaran (*reasoning*) yang tidak mengandalkan pengetahuan.(Rosnelly ; 2012 : 6).

II.1.3. Elemen Manusia Pada Sistem Pakar

Sistem pakar tidak lepas dari elemen manusia yang terkait di dalamnya. Personil yang terkait dengan sistem pakar ada 4 yaitu :

1. Pakar (*expert*)
2. Pembangun pengetahuan (*knowledge engineer*)
3. Pembangunan sistem (*system engineer*)
4. Pemakai (*user*)

Paling tidak terdapat dua komponen orang atau lebih yang berpartisipasi dalam pembangunan dan penggunaan sistem pakar, yakni sedikitnya seorang pembangun pengetahuan dan seorang pakar.(Rosnelly ; 2012 : 9).

II.1.3.1. Pakar

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu.(Rosnelly ; 2012 : 9)

II.1.3.2. Pembangun/ Pembuat Pengetahuan

Pembangun pengetahuan memiliki tugas utama menerjemahkan dan merepresentasikan pengetahuan yang diperoleh dari pakar, baik berupa pengalaman pakar dalam menyelesaikan masalah maupun sumber terdokumentasi lainnya ke dalam bentuk yang bisa diterima oleh sistem pakar. Dalam hal ini pembangunan pengetahuan (*knowledge engineer*) menginterpretasikan dan merepresentasikan pengetahuan yang diperoleh dalam bentuk jawaban-jawaban atas pertanyaan-pertanyaan yang diajukan pada pakar atau pemahaman, penggambaran analogis, sistematis, konseptual yang diperoleh dari membaca beberapa dokumen cetak seperti *text book*, jurnal, makalah, dan sebagainya. Kurangnya pengalaman *knowledgeengineer* merupakan kesulitan utama dalam mengkonstruksi sistem pakar. Untuk mengatasi hal tersebut, perancang sistem pakar menggunakan *tools* komersial. (Seperti pada editor-editor khusus maupun *logic debuggers*) dan usahanya akan dipusatkan pada pembangunan mesin inferensi.(Rosnelly ; 2012 : 10)

II.1.3.3. Pembangun/ Pembuat Sistem

Pembangun sistem adalah orang yang bertugas untuk merancang antarmuka pemakai sistem pakar, merancang pengetahuan yang sudah diterjemahkan oleh pembangun pengetahuan ke dalam bentuk yang sesuai dan dapat diterima oleh sistem pakar dan mengimplementasikannya ke dalam mesin inferensi. Selain hal tersebut, pembangun sistem juga bertanggung jawab apabila sistem pakar akan diintegrasikan dengan sistem komputerisasi lain. Alat pembangun (*tool builder*) dapat dipakai untuk menyajikan atau membangun *tool*

yang spesifik. Penjual (*vendor*) dapat memberikan *tool* dan saran, staf pendukung dapat memberikan saran dan bantuan secara teknis dalam proses pembangunan sistem pakar.(Rosnelly ; 2012 : 11)

II.1.3.4. Pengguna

Banyak sistem berbasis komputer mempunyai susunan pengguna tunggal.Hal ini berbedaa jauh dengan sistem pakar yang memungkinkan mempunyai beberapa kelas pengguna.Pengguna mungkin tidak terbiasa dengan komputer dan mungkin pada domain masalah.Bagaimanapun juga, banyak solusi permasalahan menjadi lebih baik dan kemungkinan lebih murah dan keputusan yang cepat bila menggunakan sistem pakar.Pakar dan pembangun sistem harus mengantisipasi kebutuhan-kebutuhan pengguna dan membuat batasan-batasan ketika mendesain sistem pakar.(Rosnelly ; 2012 : 11)

II.1.4. Struktur Sistem Pakar

Komponen yang terdapat dalam struktur sistem pakar ini adalah *knowledge base* (*rules*), *inference engine*, *working memory*, *explanation facility*, *knowledge acquisition facility*, *user interface*.(Rosnelly ; 2012 : 12)

1. *Knowledge Base* (Basis Pengetahuan)

   Basis pengetahuan mengandung oengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah.Komponen sistem pakar disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam

area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

2. *Inference Engine* (Mesin Inferensi)

   Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* (struktu kontrol) atau *rule interpreter* (dalam sistem pakar berbasis kaidah). Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi disini adalah *processor* pada sistem pakar yang mencocokkan bagian kondisi dari *rule* yang tersimpan di dalam *knowledge base* dengan fakta yang tersimpan di *working memory*.

3. *Working Memory*

   Berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai global *database* dari fakta yang digunakan oleh *rule-rule* yang ada.

4. *Explanation Facility*

   Menyediakan kebenaran dari solusi yang dihasilkan kepada *user* (*reasoning chain*).

5. *Knowledge Acquistion Facility*

   Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi

ke program *computer*, yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

6. *User Interface*

Mekanisme untuk memberi kesempatan kepada *user* dan sistem pakar untuk berkomunikasi.Antar muka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem.Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

II.1.5. Karakteristik Sistem Pakar

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini :

a. Kinerja sangat baik (*high performance*). Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.
b. Waktu respon yang baik (*adequate respon time*). Sistem juga harus mampu bekerja dalam waktu yang sama baiknya (*reasonable*) atau lebih cepat dibandingkan dengan seorang pakar dalam menghasilkan keputusan. Hal ini sangat penting terutama pada sistem waktu nyata (*real-time*).
c. Dapat diandalkan (*good reliability*). Sistem harus dapat diandalkan dan tidak mudah rusak/ *crash*.
d. Dapat dipahami (*understandable*). Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar.

e. Fleksibel (*flexibility*). Sistem harus menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan.(Rosnelly ; 2012 : 20).

II.2. Metode Bayes

Metode Bayes merupakan metode yang baik didalam mesin pembelajaran berdasarkan data training, dengan menggunakan probabilitas bersyarat sebagai dasarnya. Metode Bayes juga merupakan suatu metode untuk menghasilkan estimasi parameter dengan menggabungkan informasi dari sampel dan informasi lain yang telah tersedia sebelumnya. Keunggulan utama dalam penggunaan Metode Bayes adalah penyederhanaan dari cara klasik yang penuh dengan integral untuk memperoleh model marginal. Disamping itu, Metode Bayes memberikan hasil pendugaan yang lebih baik daripada pendugaan dalam metode klasik. (Pelita Informatika Budi Darma ; Sri Rahayu ; 2013 : 129).

Teorema bayes merupakan satu metode yang digunakan untuk menghitung ketidakpastian data menjadi data yang pasti dengan membandingkan antara data ya dan tidak. Probabilitas bayes merupakan salah satu cara untuk mengatasi ketidakpastian data dengan menggunakan formula bayes. (Wisnu Mahendra, dkk ; 2012).

Penelitian yang dilakukan oleh Adam danParveen (2012) menyebutkan bahwa metode Bayesdapat digunakan untuk mengembangkan SistemCerdas untuk diagnosa penyakit. Pada penelitian ini metode Bayes diimplementasikan untuk mendiagnosis penyakit jantung dan membantu praktisi kesehatan untuk membuat

keputusan klinis yang cerdas. Hasil penelitian dapat memberikan pengobatan yang efektif, dan juga membantu untuk mengurangi biaya pengobatan.

Penelitian lain ditulis oleh Sofa, dkk (2009)dengan judul Pembangunan Aplikasi Sistem Pakaruntuk Diagnosis Penyakit Tanaman Padi. Padapenelitian ini, metode *forward chaining* digunakan untuk melakukan diagnosa hama yang menyerang tanaman padi dengan melihat ciri-ciri yang muncul ditanaman tersebut. Pada beberapa kasus yang cirinyatidak ada pada aturan yang dibuat, maka dimungkinkan akan muncul hasil dimana hama yang menyerang tidak diketahui.

Penelitian lainnya yang dilakukan oleh Tuswanto dan Fadlil (2013). Penelitian ini membuat sistem pakar untuk mendiagnosa hama dan penyakit tanaman bawang merah menggunakan *certainty factor*. Penelusuran faktanya menggunakan forward chaining yaitu penelusuran yang dimulai dari fakta-fakta untuk menguji kebenaran hipotesis.

Berbeda dengan penelitian yang telah dilakukan, algoritma yang digunakan pada system pakar ini menggunakan algoritma Bayes. Dalam buku Wibisono (2009) dituliskan bahwa teori Bayes atau yang lebih dikenal dengan kaidah Bayes memainkan peranan yang sangat penting dalam penerapan probabilitas bersyarat.

Teori Bayes dikemukakan oleh seorang pendeta Inggris pada tahun 1763 yang bernama Thomas Bayes. Teori Bayes ini kemudian disempurnakan oleh Laplace. Teori Bayes digunakan untuk menghitung probabilitas terjadinya suatu peristiwa berdasarkan pengaruh yang didapat dari hasil observasi.

Teori *bayes* merupakan kaidah yang memperbaiki atau merevisi suatu probabilitas dengan cara memanfaatkan informasi tambahan. Maksudnya, dari probabilitas awal (*prior probability*) yang belum diperbaiki yang dirumuskan berdasarkan informasi yang tersedia saat ini, kemudian dibentuklah probabilitas berikutnya (*posterior probability*). (Jurnal Ilmiah Dasi ;Hartatik dan I Ketut Putra Yasa ; 2015 : 28).

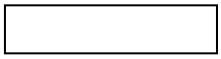
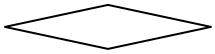
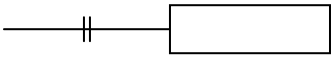
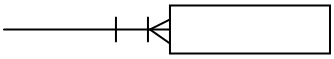
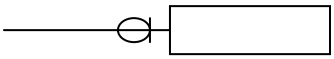
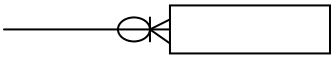
II.3. Basis Data

Basis data dapat didefenisikan sebagai koleksi dari data-data yang terorganisasi sedemikian rupa sehingga data mudah disimpan dan dimanipulasi (diperbarui, dicari, diolah dengan perhitungan-perhitungan tertentu, serta dihapus). Secara teoritis, basis data tidak harus berurusan dengan komputer (misalnya, catatan belanja hari ini yang dibuat oleh seorang ibu rumah tangga juga merupakan basis data dalam bentuk yang sangat sederhana). (Nugroho ; 2011 : 4).

II.4. *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) adalah bagian yang menunjukkan hubungan antara entity yang ada dalam sistem.Simbol-simbol yang digunakan dapat dilihat dari tabel II.6. (Jurnal Momentum ; Yuhendra, dkk ; 2015 : 70).

Tabel II.6. Simbol Yang DigunakanPada *Entity Relationship Diagram*

SIMBOL	KETERANGAN
	*Entity*
	Atribut Dan *Entity*
	Atribut Dan *Entity* Dengan *Key* (Kunci)
	Relasi Atau Aktifitas Antar *Entity*
	Hubungan Satu Dan Pasti
	Hubungan Banyak Dan Pasti
	Hubungan Satu Tapi Tidak Pasti
	Hubungan Banyak Tapi Tidak Pasti

(Sumber : Jurnal Momentum ; Yuhendra, dkk ; 2015 : 70)

II.5. Kamus Data

Kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang tegar dan teliti sehingga pemakai dan analis sistem akan memiliki pemahaman yang umum mengenai *input*, *output*, komponen penyimpanan. (Joint ; Yusi Ardi Binarso ; 2012 : 74).

II.6. Normalisasi

Normalisasi dapat dipahami sebagai tahapan-tahapan yang masing-masing berhubungan dengan bentuk normal. Bentuk normal adalah keadaan relasi yang dihasilkan dengan menerapkan aturan sederhana berkaitan dengan konsep kebergantungan fungsional pada relasi yang bersangkutan. (Nugroho ; 2011 : 199).

Kita akan menggambarkannya secara garis besar sebagai berikut :

1. Bentuk Normal Pertama (1NF/ *First Normal Form*)

   Bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak (*multivalues attribute*) telah dihilangkan sehingga kita akan menjumpai nilai tunggal (mungkin saja nilai *null*) pada perpotongan setiap baris dan kolom.

2. Bentuk Normal Kedua (2ND/ *Second Normal Form*)

   Semua kebergantungan fungsional yang bersifat sebagian (*particial functional dependency*) telah dihilangkan.

3. Bentuk Normal Ketiga (3RD/ *Thrid Normal Form*)

   Semua kebergantungan transitif (*transitive dependency*) telah dihilangkan.

4. Bentuk Normal *Boyce-Codd* (BCNF/ *Boyce-Codd Normal Form*)

   Semua anomaly yang tersisa dari hasil penyempurnaan kebergantungan fungsional sebelumnya telah dihilangkan.

5. Bentuk Normal Keempat (4NF/ *Fourth Normal Form*)

   Semua kebergantungan bernilai banyak telah dihilangkan.

6. Bentuk Normal Kelima (5NF/ *Fifth Normal Form*)

Semua anomaly yang tertinggi telah dihilangkan.(Nugroho ; 2011 : 199).

II.7. *Hypertext Preprocessor* (*PHP*)

Hypertext Preprocessor adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* banyak dipakai untuk memrogram situs *web* dynamos. *PHP* dapat digunakan untuk membangun sebuah *CMS*.Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situspersonal).PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995.Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dariweb.Selanjutnya Rasmus merilis kode sumber tersebut untuk umum danmenamakannya PHP/FI.Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP. (Jurnal Momentum ; Anisya ; 2013 : 50).

II.8. *Database MySQL*

MySQL (biasa dibaca dengan mai-es-ki-el atau bisa juga mai-se-kuel) adalah suatu perangkat lunak *database* relasi (*Relational Database Management System* atau *DBMS*), seperti halnya *ORACLE, POSTGRESQL, MSSQL,* dan lain sebagainya.*SQL* merupakan singkatan dari *Structure Query Language*, didefenisikan sebagai suatu sintaks perintah-perintah tertentu atau bahasa program yang digunakan untuk mengelola suatu *database*.Jadi *MySQL* adalah softwarenya dan *SQL* adalah bahasa perintahnya. (Jurnal Momentum ; Anisya ; 2013 : 51).

II.9. *Unified Modeling Language* (*UML*)

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (*UML*). *UML* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

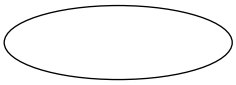
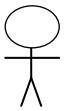
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.(Jurnal Teknologi dan Sistem Informasi ; Gellysa Urva dan Helmi Fauzi Siregar ; 2015 : 93).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan *UML* adalah sebagai berikut:

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.

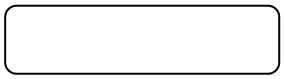
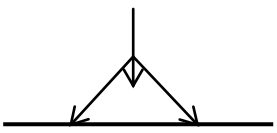
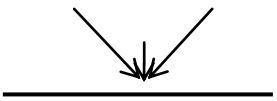
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

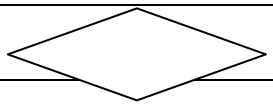
(Sumber: Jurnal Teknologi dan Sistem Informasi ; Gellysa Urva dan Helmi Fauzi Siregar ; 2015 : 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.3. Simbol *Activity Diagram*

Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.

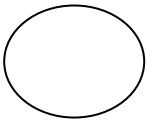
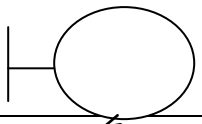
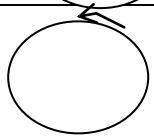
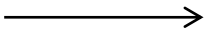
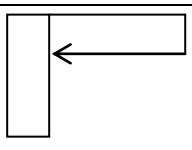
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Jurnal Teknologi dan Sistem Informasi ; Gellysa Urva dan Helmi Fauzi Siregar ; 2015 : 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	*EntityClass,* merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class,* berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*Control class,* suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive,* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber :Jurnal Teknologi dan Sistem Informasi ;Gellysa Urva dan Helmi Fauzi Siregar; 2015 : 95)

4. *Class Diagram* (Diagram Kelas)

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

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility,* tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.5 dibawah ini :

Tabel II.5. ***Multiplicity Class Diagram***

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber :Jurnal Teknologi dan Sistem Informasi ; Gellysa Urva dan Helmi Fauzi Siregar; 2015 : 95)