

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Menurut Jerry FitGerald, Ardra F. FitzGerald dan warren D. stallings, Jr., sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. (Prof. Jogiyanto HM, MBA, Akt; 2005: 2).

II.2. Pakar

Pakar adalah seseorang yang mempunyai pengetahuan dan metode khusus serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat (T.Sutojo, dkk ; 2011:163).

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu. (Rika Rosnelly; 2012; 10).

II.3. Sistem Pakar

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. Berikut adalah beberapa pengertian sistem pakar.

1. Menurut Turban (2001, p402) sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut

dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya dibutuhkan kepakaran atau keahlian manusia. (T.Sutojo, dkk; 2011: 160).

2. Prof. Edward Feignbaum dari Stanford Univercity yang merupaka pioneer sistem pakar mendefinisikan sistem pakar sebagai sebuah program komputer pintar (*intelligent computer program*) yang memanfaatkan pengetahuan (*knowledge*) dan prosedur inferensi (*inference procedure*) untuk memecahkan suatu masalah yang cukup sulit hingga membutuhkan keahlian khusus dari manusia. (Rika Rosnelly; 2012: 2).

II.3.1. Ciri-Ciri Sistem Pakar

Ciri-ciri sistem pakar antara lain:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/ *rule* tertentu
5. Mudah dimodifikasi
6. Basis Pengetahuan dan mekanisme inferensi terpisah
7. Keluarannya bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna. (T.Sutojo; 2011: 162)

II.3.2. Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikan didalamnya, diantaranya:

1. Memberikan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberikan nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal. Sistem pakar tidak pernah menjadi bosan dan kelelahan sakit.
8. Meningkatkan kapasitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional, sistem pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespons dengan: "tidak tahu" atau "tidak yakin" pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawabannya.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.

11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar. (T.Sutojo; 2011; 160-161).

II.3.3. Kekurangan sistem pakar

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya:

1. Memerlukan biaya yang sangat mahal untuk membuat dan memelihatanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem Pakar tidak selamanya 100% bernilai benar (T.Sutojo, dkk; 2011:161)

II.3.4. Karakteristik Sistem Pakar

Sistem Pakar pada umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini :

1. **Kinerja sangat baik** (*high performance*). Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau lebihhinya.
2. **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.
3. **Dapat diandalkan** (*good reliability*). Sistem harus dapat diandalkan dan tidak mudah rusak/ crash.
4. **Dapat dipahami** (*understandable*). Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar.

5. **Fleksibel** (*flexibility*). Sistem harus menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan. (Rika Rosnelly; 2012: 20-21).

II.3.5. Konsep dasar Sistem Pakar

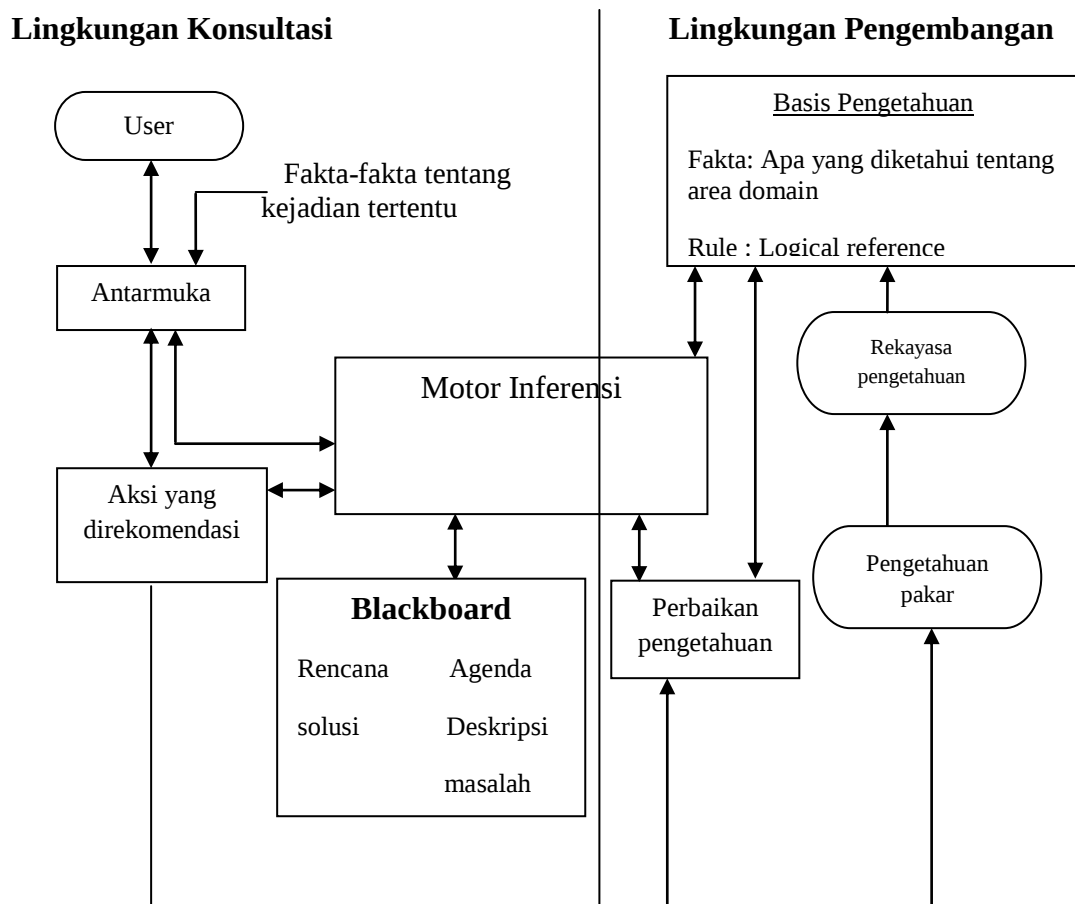
Konsep dasar sistem pakar antara lain:

1. Kepakaran (*Expertise*), yaitu suatu pengetahuan yang diperoleh dari pelatihan, membaca dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan tepat daripada seseorang yang bukan pakar.
2. Pakar (*Expert*), yaitu seseorang yang mempunyai pengetahuan, pengalaman dan metode khusus serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat.
3. Pemindahan Kepakaran (*Transferring Expertise*), yaitu proses pemindahan kepakaran dari seorang pakar ke dalam komputer kemudian mentransferkannya kepada orang lain yang bukan pakar.
4. Inferensi (*Inferencing*), yaitu sebuah prosedur (Program) yang mempunyai kemampuan dalam melakukan penalaran.
5. Aturan-aturan (*Rule*), yaitu pengetahuan disimpan dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.
6. Kemampuan Menjelaskan (*Explanation Capability*). saran atau rekomendasi yang diberikannya. Penjelasan yang dilakukan dalam subsistem yang disebut dengan (*explanatio*). (T.Sutojo, S.Si.,M.Kom, dkk; 2011; 163-165).

II.3.6. Struktur Sistem Pakar

Ada dua bagian dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan dilakukan 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. (T.Sutojo; 2011: 166).

Gambar II.1 menunjukkan komponen-komponen yang penting dalam sebuah sistem pakar.



Gambar II.1 Komponen-komponen yang penting dalam sebuah sistem pakar

(Sumber: T.Sutojo;2011: 167)

Keterangan:

1. Akusisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya 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

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* dan *backward chaining* dan gabungan dari dua 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/ Justifier*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan 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 medatang. Kemampuan evaluasi diri seperti itu diperlukan oleh program agar dapat menganalisis 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*) dari berbagai permasalahan yang ada. (T.sutojo; 2011:167-169)

II.4. Kecerdasan Buatan

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud di sini menunjuk pada mesin yang mampu berfikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia (T.Sutojo, dkk ; 2011:1).

Berikut adalah beberapa definisi kecerdasan buatan yang telah didefinisikan oleh beberapa ahli :

1. Hebert Alexander Simon (June 15, 1916 February 9, 2001) :

“Kecerdasan buatan (*Artificial Intelligence*) merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemograman komputer untuk melakukan sesuatu hal yang dalam pandangan adalah cerdas”.

2. Rich and Knight (1991) :

“Kecerdasaan buatan (AI) merupakan sebuah studi tentang bagaimana membuat computer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia”.

3. Encyclopedia Britannica

“Kecerdasan buatan (AI) merupakan cabang dari ilmu komputer yang dalam merepresentasi pengetahuan lebih banyak menggunakan symbol-simbol daripada bilangan dan memproses informasi berdasarkan metode heuristic atau dengan berdasarkan sejumlah aturan”. (T.Sutojo; 2011:1-2).

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

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. (Yuni sugiarti; 2013: 34)

II.5.1. Use Case Diagram

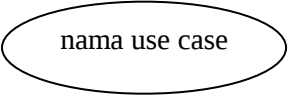
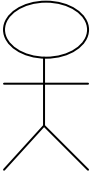
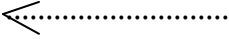
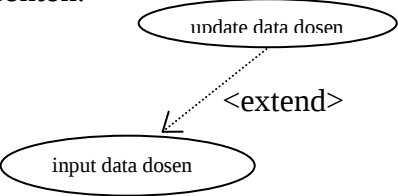
Dalam membuat sebuah sistem, langkah awal yang perlu dilakukan adalah menentukan kebutuhan. Terdapat dua jenis kebutuhan, yaitu kebutuhan *funksional* dan kebutuhan *nonfunksional*. Kebutuhan *funksional* adalah kebutuhan pengguna dan stakeholder sehari-hari yang akan dimiliki oleh sistem, dimana kebutuhan ini akan digunakan oleh pengguna dan stakeholder.

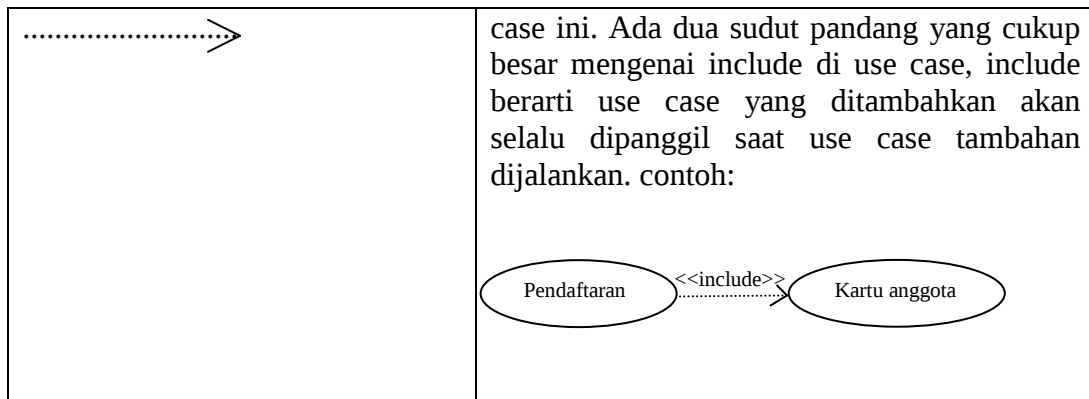
Sedangkan kebutuhan *nonfunksional* adalah kebutuhan yang memperhatikan hal-hal berikut yaitu performans, kemudahan dalam menggunakan sistem, kehandalan sistem, keamanan sistem, keuangan, legalitas dan operasional.

Kebutuhan *nonfunksional* akan digambarkan melalui sebuah diagram yang dinamakan diagram *use case*. *Use case diagram* merupakan permodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat dan

mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. (Yuni Sugiarti; 2013:41).

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 dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi dan akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Assosiaci/ accosiation 	Komunikasi antara aktor dan use case yang berpartisipasi pada use caase atau use case memiliki interaksi dengan aktor.
Extend 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapar berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case yang ditambahkan, arah panah menunjuk pada use case yang dituju. Contoh: 
Include <<include>>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use



(Sumber : Yuni Sugiarti; 2013: 42)

II.5.2. *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.
- Atribut mendeskripsikan properti dengan sebaris teks didalam kotak kelas tersebut.
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

Diagram kelas menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lainnya seperti constainment, pewarisan, asosiasi dan lain-lain.

Kelas memiliki tiga area pokok yaitu:

- a. Nama
- b. Atribut
- c. Operasi

Contoh Kelas Manusia:

- Atribut: nama, usia, tanggal lahir
- Method/ Operasi: berjalan, makan, minum (Yuni Sugiarti; 2013:57)

II.5.2.1. Abstraksi Kelas

Abstraksi adalah menentukan hal-hal mendasar pada satu objek dan mengabaikan hal-hal yang sifatnya insidental. Objek adalah instansiasi dari sebuah kelas. Abstraksi bertujuan untuk menyaring properties dan operasi pada suatu objek, sehingga hanya tinggal yang dibutuhkan saja. Sering kali masalah yang berbeda membutuhkan sejumlah informasi yang berbeda pula pada area yang sama. (Yuni Sugiarti; 2013: 58).

II.5.2.2. Atribut

Atribut adalah karakteristik data yang memiliki suatu objek kelas. (Yuni Sugiarti; 2013: 58).

II.5.2.3. Operasi

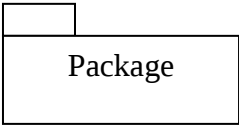
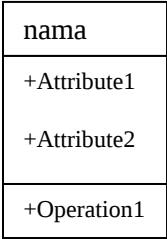
Operasi adalah fungsi atau transformasi yang mungkin dapat diaplikasikan oleh suatu objek dalam kelas. (Yuni Sugiarti; 2013: 58).

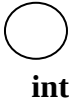
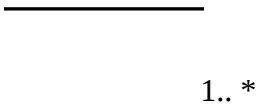
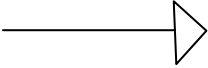
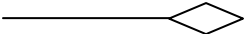
II.5.2.4. Multiplisitas/Multiplicity

Multiplisitas menunjukkan jumlah suatu objek yang bisa berhubungan dengan objek yang lain. Umumnya ditunjukkan dengan berapa banyak objek yang bisa mengisi properti “satu” atau “banyak”, tetapi secara khusus dapat ditunjukkan pula dengan bilangan integer lebih besar atau sama dengan nol.

- 1 (pasti 1)
- 0..1 (0 atau 1)
- (Tidak ada batasan, bisa 0,1, ... ,n)
- Biasanya didefinisikan batas bawah dan atas, kecuali untuk yang pasti bernilai 1. (Yuni Sugiarti; 2103: 58-59).

Tabel II.II Simbol-Simbol Diagram Kelas

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkus dari satu atau lebih kelas.
Operasi 	Kelas pada struktur sistem

Antarmuka/ Interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi Berarah/directed asosiasi 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesifikasi (umum khusus).
Kebergantungan/ defedensi 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian (whole part)

(Sumber; Yuni Sugianti; 2013 :59)

II.5.3. *Sequence Diagram*

Diagram sequence menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuance

maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diindtansikan menjadi objek itu. (Yuni Sugiarti; 2013:69)

II.5.4. Activity Diagram

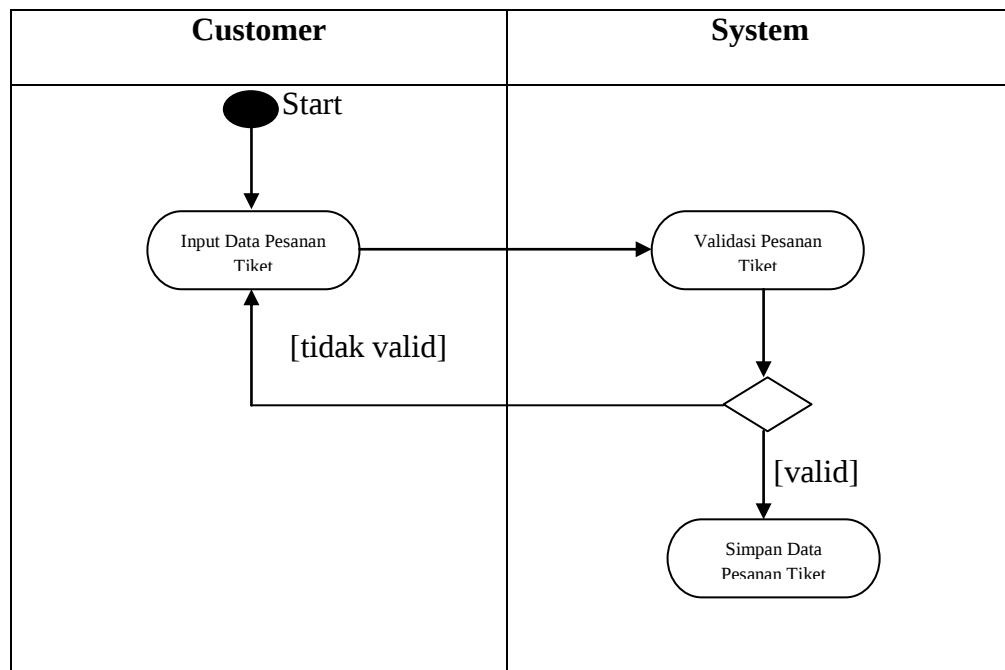
Diagram aktifitas atau *activity diagram* menggambarkan workflow (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

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 pengelompokkan 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 ujiannya. (Yuni Sugiarti;2013: 75)

Contoh *diagram activity*:

Tabel II.III Contoh Activity Diagram Entry Pesanan Tiket



(Sumber : Yuni Sugiarti; 2013: 78)

II.6. Permodelan Basis Data

Suatu basis data terdiri dari sekumpulan tabel yang saling berelasi ataupun tidak berelasi. Semua tabel tersebut merupakan representasi tempat untuk penyimpanan data yang mendukung fungsi dari basis data tersebut pada suatu sistem. Variasi kemungkinan untuk melakukan relasi yang dimiliki oleh kardinalitas terdiri dari empat macam yaitu:

- Satu ke satu (1:1)
- Satu ke banyak (1:N)
- Banyak ke satu (N:1)
- Banyak ke banyak (N:N) (Yudi priyadi; 2014: 1)

II.6.1. Model Basis data Relational

Pada awalnya, basis data *relational* diciptakan oleh seorang peneliti dari IBM yang bernama Dr. E. F. Codd, kemudian dikembangkan terus oleh peneliti lainnya hingga saat ini. Prinsipnya model basis data *relational* digunakan sebagai suatu cara untuk mengelompokkan data dari sebuah kumpulan data yang besar. Hal ini dapat dilakukan dengan cara menghapus duplikasi dari suatu data melalui proses yang disebut normalisasi. Proses ini terdiri dari beberapa langkah yang akan menjadi sebuah bentuk normal. Hasilnya berupa bahasa umum untuk melakukan akses terhadap suatu basis data, yang disebut dengan *Structure Query language*, sehingga dimungkinkan untuk melakukan *query* terhadap organisasi struktur datanya. (Yudi Pritadi; 2014:14)

II.6.2. Diagram E-R

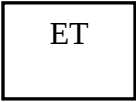
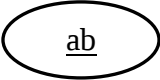
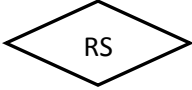
Notasi diagram E-R yang dipergunakan untuk permodelan basis data yang difokuskan pada notasi umum yang dipergunakan oleh analisis sistem, yaitu simbol entitas, relasi, atribut dan garis penghubung. Diagram tersebut akan menghasilkan permodelan atribut yang lebih banyak jumlahnya. Oleh sebab itu, kamus data (*data dictionary*) digunakan untuk melakukan penyederhanaan deklarasi atribut yang digunakan pada diagram E-R. (Yudi Priyadi; 2014:19-20)

II.6.2.1. Notasi Dasar

Permodelan basis data dengan menggunakan diagram relasi antar entitas, dapat dilakukan dengan menggunakan suatu permodelan basis data yang bernama *Diagram Entity-Relationship* (selanjutnya disingkat menjadi diagram E-R). Pada gambar di bawah ini terdapat suatu simbol/ notasi dasar yang digunakan

pada Diagram E-R, yaitu entitas, relasi, atribut dan garis penghubung. (Yudi Priyadi; 2014: 20).

Tabel II.IV Notasi Dasar E-R

Simbol	Deskripsi
	Entitas ET
	Atribut key ab
	Relasi RS
	Garis penghubung

(Sumber : Yudi Priyadi; 2014: 20)

II.7. Macromedia Dreamweaver

Dreamweaver adalah suatu bentuk program *editor web* yang dibuat oleh macromedia. Dengan program ini seorang programmer *web* dapat dengan mudah membuat dan mendesain *webnya*.

Dreamweaver adalah editor yang komplit yang dapat digunakan untuk membuat animasi sederhana yang berbentuk layer. Dengan adanya program ini kita tidak akan susah-susah membentuk script-script format HTML,PHP, ASP maupun bentuk program yang lainnya.

Sebagai editor dreamweaver mempunyai sifat yang WYSIWYG (*What You See Is What You Get*). Dengan kelebihan ini, seorang programmer dapat langsung melihat hasil buatannya tanpa harus dibuka di *browser*.

Seperti program editor-editor web lain, dreamweaver juga memiliki dua bentuk layer, yaitu bentuk halaman *design* dan halaman *code*. Hal ini akan mempermudah kita dalam menambahkan script yang berbasis PHP maupun *javascript*. Dreamweaver selain mendukung pembuatan web yang berbasis HTML, juga dapat mendukung program-program web yang lain diantaranya PHP, ASP, Perl, Javascript dan lain-lain. (Bunafit Nugroho; 2004;139)

II.8. PHP

PHP adalah bahasa pemrograman *interpreter* yang mirip dengan Bahasa C yang memiliki kesederhanaan dalam perintah. PHP dapat digunakan bersama dengan HTML sehingga memudahkan dalam pembangunan aplikasi *web* dengan cepat. PHP dapat digunakan untuk meng-*update* basis data dan menciptakan basis data. *Interpreter* adalah sebuah program yang digunakan untuk membaca *file* yang berisi kode program yang akan dijalankan, kemudian *interpreter* tersebut akan meminta CPU untuk melakukan perintah yang diterimanya.

Seperti halnya program *open source* lainnya, PHP dibuat dibawah lisensi GNU (*General Public License*), yang dapat di-*download* gratis melalui situs <http://www.php.net>. Awalnya, PHP diciptakan oleh Andi Gutsman untuk menghitung jumlah pengunjung yang mengakses *homepage* yang dibuatnya. Namun, seiring dengan perkembangan Internet, dirilis PHP/FI, PHP2, PHP3,

PHP4, dan selanjutnya PHP5. PHP5 telah mampu digunakan untuk membangun aplikasi *web* dengan koneksi basis data yang cukup banyak. PHP5 adalah versi pengembangan dari PHP4 dengan menambahkan fungsi-fungsi seperti *Zend Engine* (<http://www.zend.com>) sehingga akses lebih cepat, kuat, stabil serta mudah untuk berinteraksi dengan berbagai aplikasi pendukung lainnya.

PHP banyak mendukung basis data, seperti MySQL, PostgreSQL, Interbase, ODBC, mSQL, Oracle, dan Sybase. Kini, PHP4 banyak digunakan oleh para *web developer* untuk membangun aplikasi *web* karena memang terbukti dapat bekerja dengan baik. (Iswanto, ST; 2007; 2-3).

II.9. Metode Dempster Shafer

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 ketidakkonsistenan tersebut maka dapat menggunakan penalaran dengan teori *Dempster-Shafer*. *Dempster-Shafer* adalah suatu teori matematika untuk pembuktian berdasarkan *belief functions and plausible reasoning* (fungsi kepercayaan dan pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh Arthur P. Dempster dan Glenn Shafer. Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval :

- [Belief, Plausibility]

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. Dimana nilai bel yaitu (0-0.9).

- *Plausibility* (Pl) 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 $Pl(-s)=0$. Pada teori *Dempster-Shafer* dikenal adanya *frame of discrement* yang dinotasikan dengan θ . Frame ini merupakan semesta pembicaraan dari sekumpulan hipotesis. Tujuannya adalah mengaitkan 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 2^n . Jumlah semua m dalam subset θ sama dengan 1. Apabila tidak ada informasi apapun untuk memilih hipotesis, maka nilai : $m\{\theta\} = 1,0$.

Apabila diketahui X adalah subset dari θ , dengan m_1 sebagai fungsi densitasnya, dan Y juga merupakan subset dari θ dengan m_2 sebagai fungsi densitasnya, maka dapat dibentuk fungsi kombinasi m_1 dan m_2 sebagai m_3 , yaitu:

$$m_3(Z) = \frac{\sum_{x \cap y = z} m_1(X).m_2(Y)}{1 - \sum_{x \cap y = \emptyset} m_1(X).m_2(Y)}$$

(Sulistyohati, Aprilia dan Taufik Hidayat; 2008)

II.10. Alergi Susu Sapi

Susu sapi adalah salah satu minuman yang dikonsumsi oleh bayi. yang disebut dengan alergi susu sapi adalah kerentanan terhadap susu sapi. Sementara itu pengertian alergi itu sendiri adalah reaksi yang berlebihan terhadap stimulasi tertentu secara berulang, orang normal tidak menimbulkan reaksi. Susu sapi pada umumnya tidak menimbulkan alergi namun pada orang-orang tertentu, walaupun dalam jumlah nya sedikit, akan menimbulkan reaksi alergi baik alergi ringan maupun berat. Reaksi tersebut dapat terjadi pada setiap golongan umur, namun umumnya lebih sering terjadi pada bayi dan anak. Reaksi yang ditimbulkan dapat cepat terjadi, namun dapat juga lambat terjadinya. (Anik Maryunani; 2010:310)

II.10.1. Gejala Alergi

Manifestasi atau gejala alergi bervariasi tergantung bagian tubuh yang terkena, seperti berikut ini:

- a. Jika reaksi pada hidung: timbul rinitis dan dapat juga disertai konjungtivitis.
- b. Jika reaksi pada saluran pernafasan: asma, batuk, sesak nafas, dan bunyi mengi.
- c. Jika pada saluran pencernaan; mual, muntah, diare dan sakit perut.
- d. Jika pada kulit: eksim dan kaligata(biduran)
- e. Jika pada pembuluh darah dan sistem syaraf dapat terjadi syok.
- f. Reaksi yang terjadi dengan cepat di selaput lendir tenggorokan akan menutup jalan nafas dan menimbulkan kematian. (Anik maryunani; 2010: 310)