

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan kumpulan dari beberapa unsur atau elemen – elemen yang saling berkaitan/ berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu (Asbon Hendra; 2012 :157).

II.2. Pakar

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya. Jadi seorang pakar harus mampu melakukan kegiatan-kegiatan berikut :

1. Mengenali dan memformulasikan permasalahan.
2. Memecahkan permasalahan secara cepat dan tepat.
3. Menerangkan pemecahannya.
4. Belajar dari pengalaman.
5. Merestrukturisasi pengetahuan.
6. Memecahkan aturan-aturan. (T.Sutojo, dkk ; 2011 : 163-164)

II.3. Sistem Pakar

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah (T.Sutojo, dkk ; 2011 : 13).

Menurut Turban (2001, p402), sistem pakar adalah sebuah system yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah – masalah yang biasanya membutuhkan kepakaran atau keahlian manusia (T.Sutojo, dkk; 2011: 160).

II.3.1. Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, di antaranya :

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi 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 atau sakit.

8. 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.
9. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar (T.Sutojo, dkk ; 2011 : 160-161).

II.3.2. Kekurangan Sistem Pakar

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

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

II.3.3. Pemindahan Kepakaran

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransferkan kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu :

1. Akuisisi pengetahuan (dari pakar atau sumber lain)
2. Representasi pengetahuan (pada komputer)
3. Inferensi pengetahuan
4. Pemindahan pengetahuan ke pengguna (T.Sutojo, dkk ; 2011 : 164)

II.4. Gunung

Gunung adalah sebuah bentuk tanah yang menonjol di atas wilayah sekitarnya. Sebuah gunung biasanya lebih tinggi dan curam dari sebuah bukit, tetapi ada kesamaan, dan penggunaan sering tergantung dari adat lokal. Beberapa otoritas mendefinisikan gunung dengan puncak lebih dari besaran tertentu. (<http://id.wikipedia.org/wiki/Gunung>)

II.4.1. Gunung Berapi

Gunung berapi adalah istilah yang dapat didefinisikan sebagai suatu sistem saluran fluida panas (batuan dalam wujud cair atau lava) yang memanjang dari kedalaman sekitar 10 km di bawah permukaan bumi sampai ke permukaan bumi, termasuk endapan hasil akumulasi material yang dikeluarkan pada saat meletus. ([http://id.wikipedia.org/wiki/Gunung Berapi](http://id.wikipedia.org/wiki/Gunung_Berapi))

II.4.2. Jenis Gunung Berapi Berdasarkan Bentuk

Gunung Berapi yang terdapat di dunia dapat di klasifikasikan menjadi 4 bentuk umum,yaitu;

1. Stratovolcano

Tersusun dari batuan hasil letusan dengan tipe letusan berubah-ubah sehingga dapat menghasilkan susunan yang berlapis-lapis dari beberapa jenis batuan, sehingga membentuk suatu kerucut besar (raksasa), kadang-kadang bentuknya tidak beraturan, karena letusan terjadi sudah beberapa ratus kali. Gunung Merapi merupakan jenis ini.

2. Perisai

Tersusun dari batuan aliran lava yang pada saat diendapkan masih cair, sehingga tidak sempat membentuk suatu kerucut yang tinggi (curam), bentuknya akan berlereng landai, dan susunannya terdiri dari batuan yang bersifat basaltik. Contoh bentuk gunung berapi ini terdapat di kepulauan Hawai.

3. Cinder Cone

Merupakan gunung berapi yang abu dan pecahan kecil batuan vulkanik menyebar di sekeliling gunung. Sebagian besar gunung jenis ini membentuk mangkuk di puncaknya. Jarang yang tingginya di atas 500 meter dari tanah di sekitarnya.

4. Kaldera

Gunung berapi jenis ini terbentuk dari ledakan yang sangat kuat yang melempar ujung atas gunung sehingga membentuk cekungan. Gunung Bromo merupakan jenis ini. (http://id.wikipedia.org/wiki/Gunung_Berapi)

II.4.3. Status Gunung Berapi

Status Gunung berapi merupakan tingkat isyarat dalam penentuan status gunung tersebut dan menentukan tindakan yang akan di lakukan masyarakat sekitar dalam menghadapi status gunung yang terjadi.

Tabel II.1. Status Gunung Berapi

Tingkat isyarat gunung berapi di Indonesia		
Status	Makna	Tindakan
AWAS	- Menandakan gunung berapi yang segera atau sedang meletus atau ada keadaan kritis yang menimbulkan bencana - Letusan pembukaan dimulai dengan abu dan asap - Letusan berpeluang terjadi dalam waktu 24 jam	- Wilayah yang terancam bahaya direkomendasikan untuk dikosongkan - Koordinasi dilakukan secara harian - Piket penuh
SIAGA	- Menandakan gunung berapi yang sedang bergerak ke arah letusan atau menimbulkan bencana - Peningkatan intensif kegiatan seismik - Semua data menunjukkan bahwa aktivitas dapat segera berlanjut ke letusan atau menuju pada keadaan yang dapat menimbulkan bencana - Jika tren peningkatan berlanjut, letusan dapat terjadi dalam waktu 2 minggu	- Sosialisasi di wilayah terancam - Penyiapan sarana darurat - Koordinasi harian - Piket penuh
WASPADA	- Ada aktivitas apa pun bentuknya - Terdapat kenaikan aktivitas di atas level normal - Peningkatan aktivitas seismik dan kejadian vulkanis lainnya - Sedikit perubahan aktivitas yang diakibatkan oleh	- Penyuluhan/sosialisasi - Penilaian bahaya - Pengecekan sarana - Pelaksanaan piket terbatas

	aktivitas magma, tektonik dan hidrotermal	
NORMAL	• Tidak ada gejala aktivitas tekanan magma • Level aktivitas dasar	• Pengamatan rutin • Survei dan penyelidikan

([http://id.wikipedia.org/wiki/Gunung Berapi](http://id.wikipedia.org/wiki/Gunung_Berapi))

II.5. Certainty Factor

Teori *Certainty Factor* diusulkan oleh Shortliffe dan Buchanan pada 1975 untuk mengakomodasi ketidakpastian pemikiran (*inexact reasoning*) seorang pakar. Seorang pakar, misalnya dokter sering kali menganalisis informasi yang ada dengan ungkapan seperti “mungkin”, “kemungkinan besar”, “hampir pasti”. Untuk mengakomodasi hal ini perlu menggunakan *Certainty Factor* untuk menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.

Ada dua cara dalam mendapatkan tingkat keyakinan dari sebuah rule, yaitu:

1. Metode ‘*Net Belief*’ yang diusulkan oleh E.H. Shortliffe dan B.G. Buchanan

$$CF(\text{Rule}) = MB(H,E) - MD(H,E)$$

$$MB(H,E) = \begin{cases} 1 & P(H) = 1 \\ \frac{\max[P(H|E), P(H)] - P(H)}{\max[1,0] - P(H)} & \text{otherwise} \end{cases}$$

$$MD(H,E) = \begin{cases} 1 & P(H) = 0 \\ \frac{\min[P(H|E), P(H)] - P(H)}{\max[1,0] - P(H)} & \text{otherwise} \end{cases}$$

Keterangan :

- a. $CF(Rule)$ = faktor kepastian
- b. $MB(H,E)$ = *measure of belief* (ukuran kepercayaan) terhadap Hipotesis H, jika diberikan *evidence* E (antara 0 dan 1)
- c. $P(H)$ = probabilitas kebenaran hipotesis H
- d. $P(H|E)$ = probabilitas bahwa H benar karena fakta E
(T.Sutojo,dkk;2011:194)

II.5.2. Kelebihan dan kekurangan Metode Certainty Factor

Kelebihan dari metode *Certainty Factor* adalah:

1. Metode ini cocok dipakai dalam sistem pakar yang mengandung ketidakpastian.
2. Dalam sekali proses perhitungan hanya dapat mengolah 2 data saja sehingga keakuratan data dapat terjaga.

Sedangkan kekurangan metode *Certainty Factor* adalah:

3. Pemodelan ketidakpastian yang menggunakan perhitungan metode *Certainty Factor* biasanya masih diperdebatkan.
4. Untuk data lebih dari 2 buah, harus dilakukan beberapa kali pengolahan data (T. Sutojo, dkk; 2011: 204).

II.6. PHP

Menurut kamus komputer, PHP adalah bahasa pemrograman untuk dijalankan melalui halaman web, umumnya digunakan untuk mengolah informasi di internet. Sedangkan dalam pengertian lain, PHP adalah singkatan dari *PHP Hypertext Preprocessor* yaitu bahasa pemrograman web *server-side* yang bersifat *open source* atau gratis. PHP merupakan *script* yang menyatu dengan HTML dan berada pada server (*server side HTML embedded scripting*) (Rulianto Kurniawan; 2010: 2).

II.7. MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal. MySQL termasuk jenis RDBMS (*Relational Database Management System*). MySQL ini mendukung bahasa pemrograman PHP. MySQL juga mempunyai *query* atau bahasa SQL (*Structured Query Language*) yang simpel dan menggunakan *escape character* yang sama dengan PHP (Rulianto Kurniawan; 2010: 16).

II.8. UML (Unified Modelling Language)

UML (*Unified Modeling Language*) adalah sebuah ”bahasa” yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Seperti bahasa-bahasa lainnya, UML mendefenisikan notasi dan *syntax/semantic*. Notasi UML merupakan sekumpulan

bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk – bentuk tersebut dapat dikombinasikan.

Unified Modeling Language biasa digunakan untuk :

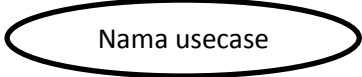
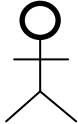
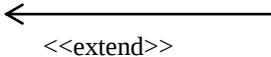
1. Menggambarkan batasan sistem dan fungsi – fungsi sistem secara umum, di buat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang di laksanakan secara umum, di buat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*.

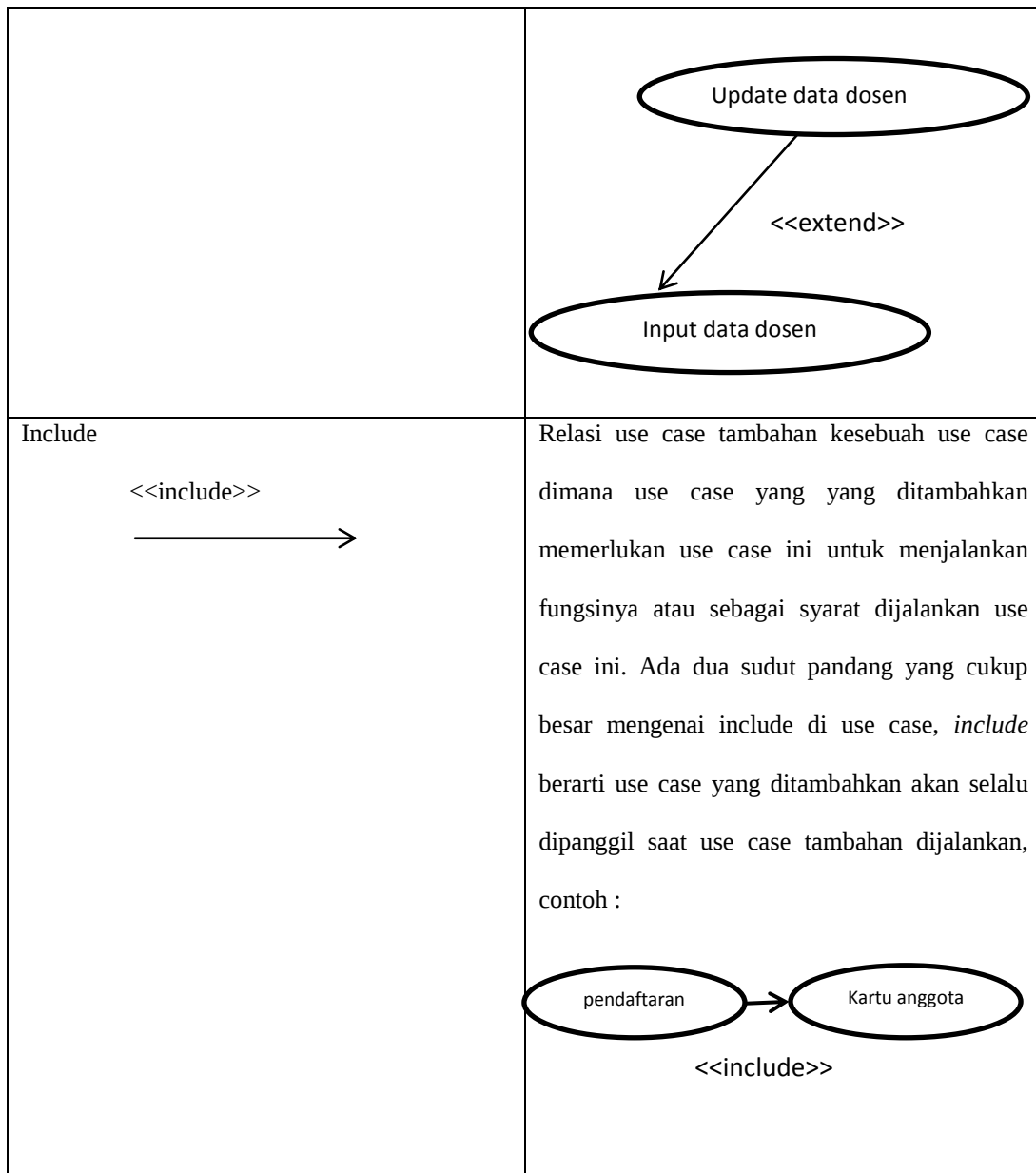
II.8.1. Use Case Diagram

Use case diagrams merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. Terdapat beberapa simbol dalam

menggambarkan diagram *use case*, yaitu *use case*, *actor* dan relasi. Berikut adalah simbol – simbol yang ada pada diagram *use case* :

Tabel II.2 Simbol – simbol Pada Use Case

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau <i>actor</i> ; biasanya ditanyakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
Aktor  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i> .
Asosiasi/ <i>association</i> 	Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
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 ditambahkan, arah panah menunjuk pada use case yang dituju. Contoh :



Sumber: (Yuni Sugiarti; 2013,42)

II.8.2. Class Diagram

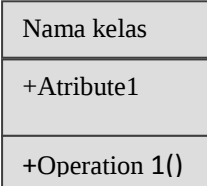
Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefenisian kelas – kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang di sebut atribut dan metode atau operasi.

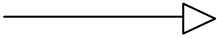
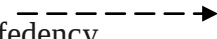
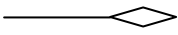
1. Atribut merupakan variabel- variabel yang di miliki oleh suatu kelas.

2. Atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut.
3. Operasi atau metode adalah fungsi – fungsi yang di miliki 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.

Tabel II.3 Simbol – simbol Class Diagram

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem
Antarmuka / interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah/directed asosiasi 	Relasi antar kelas dengan makna kelas yang satu di gunakan oleh kelas yang lain, asosiasi biasanya juga di sertai dengan <i>multiplicity</i> .

Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus).
Kebergantungan / defedency 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Sumber: (Yuni Sugiarti; 2013,59)

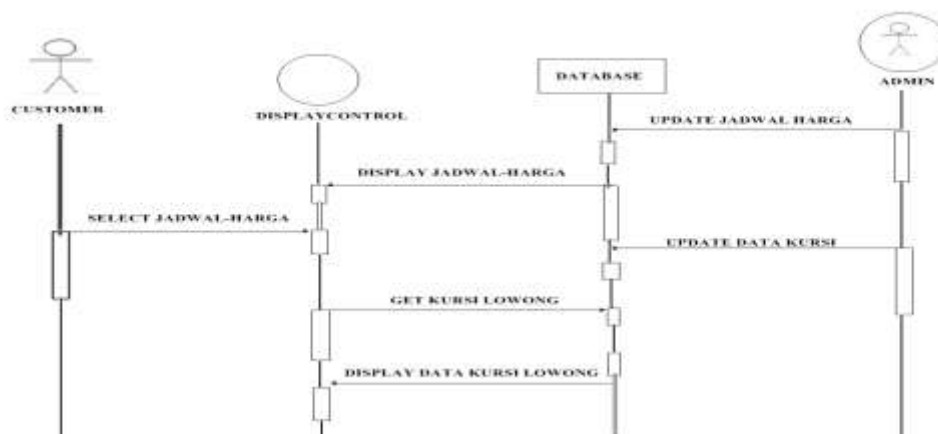
II.8.3. Sequence Diagram

Diagram sekueunce menggambarkan kelakuan/ pelaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek – objek yang terlibat dalam sebuah use case beserta metode – metode yang dimiliki kelas yang diinstasiasi menjadi objek itu.

Diagram sequence memiliki ciri yang berbeda dengan diagram interaksi pada diagram kolaborasi sebagai berikut :

1. Pada diagram sequence terdapat garis hidup objek. Garis hidup objek adalah garis vertical yang mencerminkan eksistensi sebuah objek sepanjang periode waktu. Sebagian besar objek – objek yang tercakup dalam diagram interaksi akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek – objek itu diletakkan dibagian atas diagram dengan garis hidup tergambar dari atas hingga bagian bawah diagram. Suatu objek lain dapat saja diciptakan, dalam hal ini garis hidup dimulai saat pesan *destroy*, jika kasus ini terjadi, maka garis hidupnya juga berakhir.

2. Terdapat focus kendali (*Focus Of Control*), berupa empat persegi panjang ramping dan tinggi yang menampilkan aksi suatu objek secara langsung atau sepanjang sub ordinat. Puncak dari empat persegi panjang adalah permulaan aksi, bagian dasar adalah akhir dari suatu aksi. Pada diagram ini mungkin juga memperhatikan penyaringan (*nesting*) dan *focus* kendali yang disebabkan oleh proses rekursif dengan menumpuk *focus* kendali yang lain pada induknya. Keterangan gambar di kutip pada



Gambar II.I: Contoh Sequence Diagram
(Sumber: Yuni Sugiarti,2013,71)

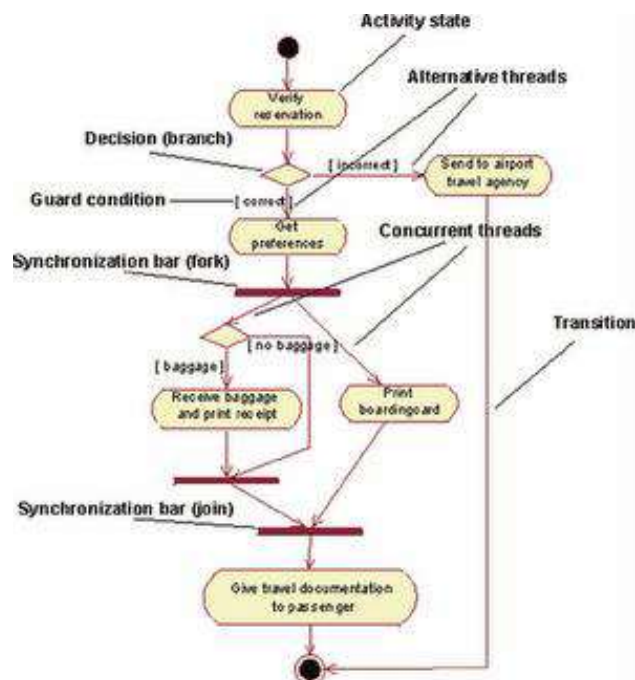
II.8.4. Activity Diagram (Aktivitas)

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem)

secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.



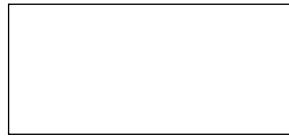
Gambar II.2: Contoh Activity Diagram
(Sumber: Jurnal, Memahami Penggunaan UML,2011)

II.9. ERD (Entity Relationship Diagram)

Entity relationship diagram adalah pemodelan basis data yang fokusnya pada notasi yang umum di pergunakan oleh analisis sistem,yaitu simbol entitas,relasi dan garis penghubung. Elemen – elemen diagram hubungan entitas yaitu

1. Entitas (*Entity*)

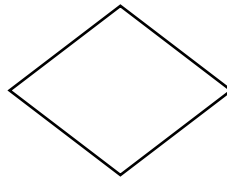
Entitas merupakan notasi untuk mewakili suatu objek dengan karakteristik sama, yang dilengkapi dengan atribut, sehingga pada suatu lingkungan nyata setiap objek akan berbeda dengan objek lainnya. Pada umumnya, objek dapat berupa benda, pekerjaan dan orang.



Gambar II.3: Simbol *Entitas*
(Sumber : Yudi Priadi, 2013, 20)

2. Relasi (*Relationship*)

Relasi adalah notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.



Gambar II.4: Simbol *Relationship*
(Sumber : Yudi Priadi, 2013, 20)

3. Atribut (*Attribute*)

Atribut adalah notasi yang menjelaskan karakteristik suatu entitas dan juga relasinya. Atribut dapat sebagai key yang bersifat unik, yaitu primary key dan foreign key.



Gambar II.5: Simbol *Attribute*
(Sumber : Yudi Priadi, 2013, 20)

4. Garis Penghubung adalah untuk mengkaitkan keterkaitan antara notasi notasi yang di gunakan dalam Diagram E-R, yaitu:itu entitas,relasi dan atribut.

Gambar II.6: Simbol Garis Penghubung

(Sumber : Yudi Priadi,2013,20)

II.9.1 Normalisasi

Normalisasi adalah bagian perancangan basis data. Tanpa normalisasi, sistem basis data menjadi tidak akurat, lambat, tidak efisien, serta tidak memberikan data yang diharapkan. Pada waktu menormalisasikan basis data, ada empat tujuan yang harus dicapai,.

1. Mengatur data dalam kelompok – kelompok sehingga masing – masing kelompok hanya mengenai bagian kecil sistem.
2. Meminimalkan jumlah data berulang dalam basis data.
3. Membuat basis data yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan integritas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada satu tempat.

II.10. Kamus Data

Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan

keluaran (*ouput*) dapat dipahami secara umum (memiliki standar cara penulisan).

Kamus data biasanya berisi :

1. Nama – nama dari data.
2. Digunakan pada – merupakan proses – proses yang terkait data.
3. Deskripsi – merupakan deskripsi data
4. Informasi tambahan, seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.