BAB II

LANDASAN TEORI

II.1. Kecerdasan Buatan

Kecerdasan buatan adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berprilaku cerdas seperti manusia.Ilmu komputer tersebut mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia. Aktifitas manusia yang ditirukan seperti penalaran,penglihatan,pembelajaran, pemecahan masalah, pemahaman bahasa alami dan sebagainya.

Sesuai dengan definisi tersebut maka teknologi kecerdasan buatan dipelajari dalam bidang seperti : Robotika (*Robotics*),Penglihatan komputer *(Computer Vision),*Pengolahan bahasa alami (*Natural Language Prossesing*),Pengenalan pola (*Pattern Recognition*), Sistem Syaraf Buatan (*Artificial Neural System*),Pengenalan Suara (*SpeechRecognition*), dan sistem Pakar (*Expert System*).Kecerdasan buatan menyelesaikan permasalahan dengan mendayagunakan komputer untuk memecahkan masalah yang komplek dengan cara mengikuti proses penalaran manusia.Salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia adalah sistem pakar.(Yani Immarul Za’iim; 2013:2).

II.1.1. Tipe Pengetahuan dalam Penjelasan Sistem Pakar

Secara khusus ada tiga tipe pengetahuan yang akan dibahas yaitu :

1. ***Resoning Domain Knowledge*** (RDK), merupakan domain pengetahuan yang dikodekan oleh domain pakar dalam sistem pakar yang sesuai.
2. ***Communication Domain Knowledge (CDK),*** merupakan pengetahuan tentang doamain pakar yang diperlukan untuk komunikasi tentang domain itu.
3. ***Domain Communication Knowledge (DCK),*** merupakan pengetahuan tentang bagaimana cara mengkomunikasikan doamain itu.(Rika Rosnelly; 2012:108)

II.2. Sistem Pakar

Bidang sistem pakar merupakan penyelesaian pendekatan yang sangat berhasil dan bagus untuk AI klasik dari pemograman *intelligent* (cerdas). Sistem pakar (*expert sistem)* merupakan solusi AI bagi masalah pemograman pintar. Professor Edward Feigenbaum dari Stanford University yang merupakan pionir dalam teknologi sistem pakar mendefenisikan sistem pakar sebagai sebuah program komputer pintar (*intelligent computer program)* yang memanfaatkan pengetahuan (*knowledge)* dan prosedur inferensi (*inference procedure)* untuk memecahkan masalah yang cukup sulit sehingga membutuhkan keahlian khusus dari manusia. (Rika Rosnelly; 2012; 2)

Sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek *(emulates)* kemampuan pengambilan keputusan *(decision making)* seorang

pakar.Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah. (Rika Rosnelly; 2012:2)

II.2.1. Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya adalah :

1. Meningkatkan ketersediaan.
2. Mengurangi biaya.
3. Mengurangi bahaya.
4. 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.
5. Keahlian multiple.
6. Meningkatkan kehandalan *(increased reliability).*
7. Respon yang cepat *(Fast response).*
8. Stabil,tidak emosional,dan memberikan respon yang lengkap setiap saat *(steady,unemotional,and completeresponse at alltimes).*
9. Pembimbing pintar *(intelligent tutor).*
10. Basis data cerdas *(intellegent database).* Sistem pakar dapat digunakan untuk mengakseskan basis data secara cerdas. (Rika Rosnelly; 2012: 5)

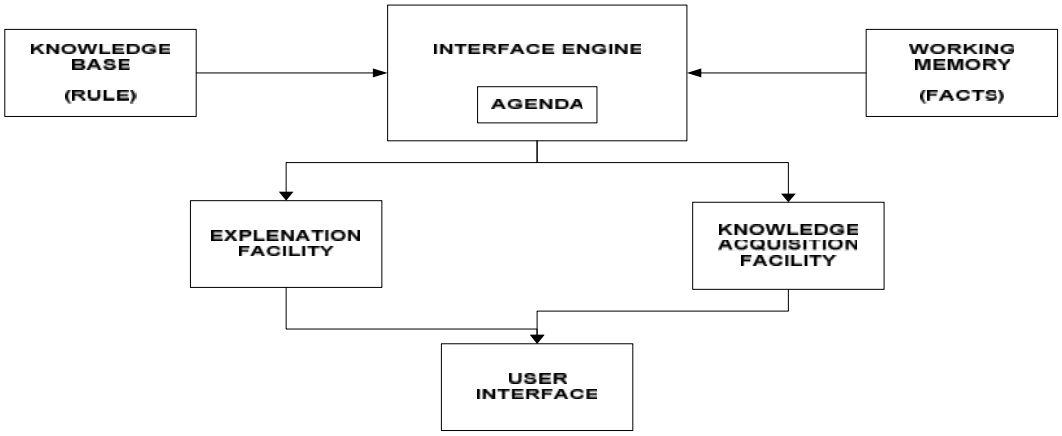
II.2.2. Karakteristik Sistem Pakar

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

1. Kinerja sangat baik (*high performance),*Sistem harus mampu memberikan respon berupa saran *(advice)* Waktu respon yang baik *(adequate respon time).*Sistem juga harus mampu bekerja dalam waktu yang sama.
2. Dapat diandalkan.
3. Dapat dipahami.
4. *Fleksibel.* (Rika Rosnelly;2012:20)

II.2.3. Struktur Sistem Pakar

Adapun struktur sistem pakar dapat dilihat pada Gambar II.1.



Gambar II.1 : Struktur Sistem Pakar
Sumber : (Rika Rosnelly; 2012:13)

Komponen yang terdapat pada dalam struktur sistem pakar ini adalah *knowledge base (rules), inference engine, working memory, explanation facility, knowledge acquisition facility, dan user interface.* (Rika Rosnelly; 2012: 13)

II.2.4. Pakar

Pakar adalah seorang individu yang memiliki pengetahun khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertetu.

Seorang pakar memiliki kemampuan kepakaran, yaitu :

1. Dapat mengenali dan merumuskan suatu masalah.
2. Menyelesaikan masalah dengan cepat dan tepat.
3. Menjelaskan solusi dari suatu masalah.
4. Restrukturisasi pengetahuan.
5. Belajar dari pengalaman.
6. Memahami batas kemampuan. (Rika Rosnelly; 2012: 10)

II.3.Perceptron

Secara umum teori Perceptron termasuk salah satu bentuk Jaringan Syaraf Tiruan *(Neural Network)* yang sederhana. Pada dasarnya *perceptron* pada Jaringan Syaraf Tiruan *(Neural Network)* dengan satu lapisan memiliki bobot yang bisa diatur dan suatu nilai ambang.Algoritma yang digunakan oleh aturan *perceptron* ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Fungsi aktivasi dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif. untuk mengontrol perubahan bobot pada setiap iterasi, Besarnya nilai lebih besar dari 0 (nol) dan maksimal Besarnya perubahan bobot yang terjadi pada setiap iterasi adalah:

Adapun Rumus Metode Perceptron berikut ini :

$$Y\_in = \sum Xi.Wi$$

Keterangan : input y_in pada neuron Y merupakan penjumlahan dari perkalian neuron-neuron input dengan masing-masing bobot yang bersesuaian.lalu di berikan fungsi aktivasi untuk menghasilkan neuron Y

$$Y = \sum_{i=0}^{n} Xi.Wi$$

Keterangan : W*i* = nilai bobot

x*i*= masukan/inputan

Penjumlahan bobot yang berada di atas atau di bawah nilai ambang (threshold) yang telah ditentukan dengan aturannya sebagai berikut :

Menentukan hasil Aktivasi dan target error :

y = {0, Jika x<0,5

{1, Jika x>= 0,5

Error : (t-y)

Dimana : t = nilai keluaran (output)

Keluaran ini kemudian dibandingkan dengan hasil (target) yang diinginkan.sehingga dihasilkan keluaran yang sesuai dengan hasil yang diinginkan dengan rumus perubahan bobotnya :

$$W_i \text{ (baru)} = W_i \text{ (Lama)} + a \text{ (t-y) } X_i$$

Keterangan :

a = Kecepatan belajar/*learning rate*

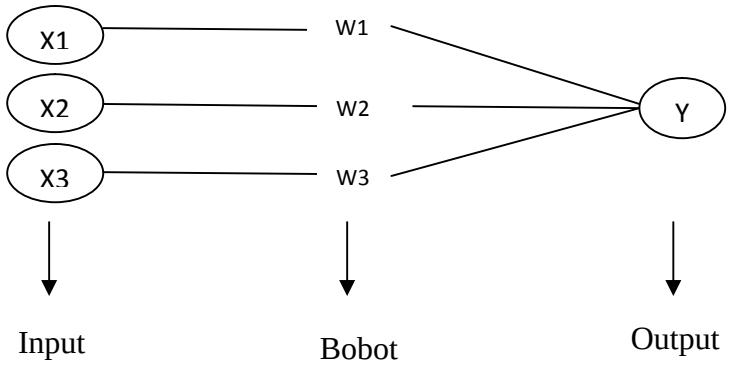
t = nilai keluaran

y = nilai aktivasi.

xi = masukan/inputan yang sudah ditentukan. (Diana Laily Fithri;2011:5)

II.4. **Jaringan Saraf Tiruan Perceptron**

Perceptron pada Jaringan Syaraf Tiruan *(Neural Network)* termasuk kedalam salah satu bentuk Jaringan Syaraf *(Neural Network)* yang sederhana.Pada dasarnya *perceptron* pada Jaringan Syaraf Tiruan *(Neural Network)* dengan lapisan memiliki bobot/nilai range yang bisa diatur dan suatu nilai ambang.Algoritma yang digunakan oleh aturan perceptron ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran.Fungsi aktivasi dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif.Didalam JST penggunaanya dapat terbagi menjadi bermacam Arsitektur dan JST sederhana contonya :



Proses Langkah dari perceptron adalah sebagai berikut :

1. Inisialisasi semua bobot dan set learning rate $\alpha(0 < \alpha <= 1)$.untuk penyederhanaan set sama dengan 1 set nilai threshold (θ) untuk fungsi Aktivasi.
2. Jaringan Input y_in pada Neuron Y merupakan penjumlahan dari perkalian Neuron input dimana Neuron Y menerima input dari neuron x1,x2,x3 serta w1,w2,w3 dengan masing-masing bobot dimana : Y_in ∑xi.wi
3. Lalu Y_in diberikan fungsi Aktivasi untuk menghasilkan Neuron Y dimana :

$$Y = \sum_{i=0}^{n} xi.wi$$

4. Masukkan kedalam fungsi Aktivasi : y = {0, Jika x<0,5

   {1, Jika x>=0,5

5. Kemudian bandingkan dengan hasil (target) yang diinginkan sehingga dihasilkan keluaran yang sesuai dengan hasil yang diinginkan dengan cara berikut : wi(baru) = wi(lama) + a(t-y)xi
6. Setelah semua sudah ditentukan maka munculah hasil bobot baru yang baik atau sesuai. (Yani Immarul Za'iim;2013:3).

II.4.1. Kulit

Kesehatan kulit perlu diperhatikan karena kulit merupakan jaringan/organ yang paling vital serta cermin dari kesehatan dan kehidupan manusia.Fungsi kulit adalah melindungi tubuh terhadap serangan penyakit dari luar dan menjaga suhu tubuh agar tetap normal.Selain itu, kulit juga memiliki nilai estetika.Gangguan pada kulit sering terjadi karena berbagai faktor antara lain : iklim,lingkungan tempat tinggal,kebiasaan hidup yang kurang sehat,alergi,dll. Hambatan-hambatan yang menyebabkan sulitnya melakukan konsultasi penyakit kulit sekarang ini dapat diatasi dengan adanya program komputer. Dalam hal ini sistem pakar dapat membantu pemecahan masalah terhadap penyakit kulit dengan di berikan nasihat kepada pemakai dan menemukan solusi terhadap berbagai permasalahan yang spesifik .(Tanda Selamat;2011:1)

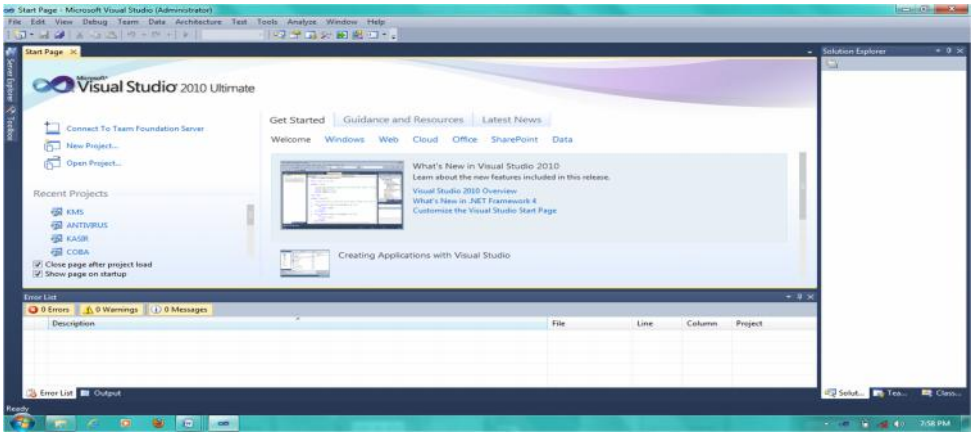
II.5. Penyakit kulit (*Acne Vulgaris*)

Acne vulgaris merupakan penyakit *self limited* (sembuh sendiri) pada unit pilosebaseus terutama terjadi pada orang dewasa yang ditandai dengan adanya lesi *pleomorfik* seperti komedo, papul, pustul, nodus, kista, dan jaringan parut,baik yang *hipertrofik* maupun yang *hipotrofik* dengan predileksi di wajah, dada, punggung, dan bahu. Patogenesis *acne vulgaris* terdiri dari empat faktor yang saling mempengaruhi, yaitu *hiperkeratinisasi folikuler,kolonisasi* bakteri *Propionibacterium acnes*, peningkatan produksi sebum, dan inflamasi.Berbagai macam terapi sistemik seperti antibiotik telah digunakan untuk pengobatan *Acne Vulgaris* dengan tujuan menurunkan jumlah *P. acnes* dan sebagai anti inflamasi. Secara in vitro, *P. acnes* sangat sensitif terhadap beberapa antibiotik dari golongan

yang berbeda, termasuk makrolida, tetrasiklin,penisilin,klinda-misin, sefalosporin, trimetoprin, dan sulfonamid. (Satya Wyda Yenny ; 2011:169).

II.6. Pemrograman Visual Basic 2010

Visual basic 2010 merupakan salah satu bagian dari produk pemograman terbaru yang dikeluarkan oleh Microsoft, sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan Microsoft. Visual Studio merupakan produk pemograman andalan dari Microsoft Corporation, yang didalamnya berisi beberap IDE pemograman seperti Visual Basic, Visual C++, Visual Web Developer, Visual C# dan Visual F#. Semua IDE tersebut sudah mendukung penuh implementasi.Visual Basic 2010 merupakan versi pebaikan dan pengembangan dari versi pendahuluannya yaitu Visual Basic 2008. Dari Pengembangan di dalamnya ada dukungan terhadap pengembangan aplikasi menggunakan *Microsoft SilverLight*,dukungan terhadap aplikasi berbasis *Cloud Computing*,serta perluasan dukungan terhadap database-database baik *standalone* maupun *database* server.



Gambar II.2 : Microsoft Visual Basic 2010

(Sumber : Wahana Komputer; 2010: 3)

II.7. SQL Server 2008

SQL Server 2008 adalah sebuah DBMS *(Database Management System)* yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. (Andi Gunawan ; 2011:2)

II.8. Pengertian Database

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk.* Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database*.

Pengaplikasian *database* dapat kita lihat dan rasakan dalam keseharian kita. *Database* ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM *(*anjungan tunai mandiri / *automatic teller machine*) bank karena banktelah mempunyai *database* tentang nasabah dan rekening nasabah.Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks *database* sebenarnya kita sudah melakukan perubahan *(update)*

data pada *database* di bank.Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep *database*. Data yang kita simpan di HP juga mempunyai struktur yang diisi melalui formulir *(form)* yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya. (Agustinus Mujilan ; 2012 : 23).

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

UML sesuai dengan kata terakhir dari kepanjangannya, UML itu adalah salah satu bentuk *language* atau bahasa. Menurut pencetusnya, UML didefenisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem. Karena tergolong bahasa visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan. (Yuni Sugiarti; 2013: 34)

Unified Modelling Language (UML) biasanya digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum.
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagram*.
4. Membuat model *behavior* “yang menggambarkan sifat atau sebuah sistem.

5. Menyatakan arsitektur implementasi fisik menggunakan *component* dan *development diagram.*
6. Menyampaikan atau memperluas *fungsionalty* dengan *stereotypes.* (Yuni Sugiarti; 2013: 34)

II.10. Jenis-Jenis Diagram UML

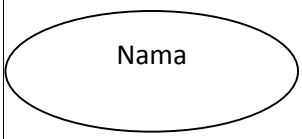
Berikut ini adalah mengenai berbagai diagram UML serta tujuan dan simbol-simbolnya, yaitu:

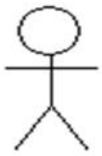
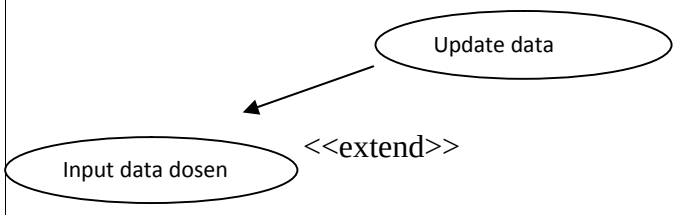
1. *Use Case Diagram* (Diagram *Use Case*)

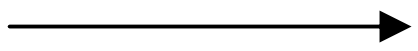
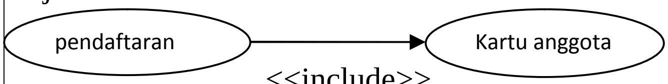
   Use case diagram secara garis besar menggambarkan interaksi antara sistem, sistem ekstrenal, dan pengguna. Dengan kata lain secara garis besar *use case diagram* secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna *(user)* mengharapkan interaksi dengan sistem itu. Terdapat beberapa simbol dalam menggambarkan diagram *use case*, yaitu *use case*, actor, relasi.

   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  Nama aktor	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 aktor
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 bambahkan, arah panah menunjukkan pada *use case* yang dituju.contoh: 

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 use case, include berarti *use case* yang ditambahkan akan selalu di panggil saat *use case* tambahan dijalankan. Contoh: 

Sumber: (Yuni Sugiarti; 2013: 42)

2. *Class Diagram* (Diagram Kelas)

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

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

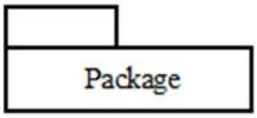
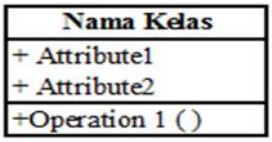
Kelas memiliki tiga area pokok :

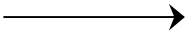
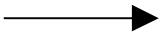
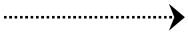
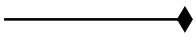
1. Nama
2. Atribut
3. Operasi

Contoh kelas “Manusia”, Atribut: nama, manusia, tanggal lahir, Method/operasi: berjalan, makan, minum

Berikut ini adalah simbol-simbol yang ada pada *class* diagram :

Tabel II.2 Simbol-Simbol *Class Diagram*

Simbol	Deskripsi
Package 	Package merupakan bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem
Antarmuka/interface  interface	Sama dengan konsep *interface* dalam pemograman berorientasi objek
Asosiasi 1 1..*	Relasi antar kelas dengan maknaumum, asosiasi biasanya juga disertai dengan *multiplicty*
Asosiasi berarah / directed asosiasi	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicty*

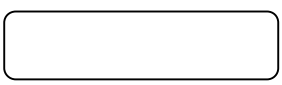
	
Generalisasi 	Relasi antar kelas dengan makna *generalisasi-spesialisasi* (umum-khusus)
Kebergantungan (*defedency)* 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi 	Relasi antar kelas dengan makna semua-bagian (*whole-part*)

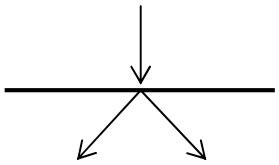
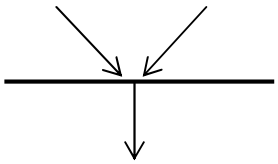
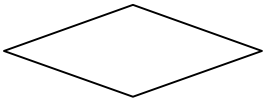
Sumber: (Yuni Sugiarti; 2013: 59)

3. *Actifity Diagram* (Diagram Aktivitas)

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.3 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.

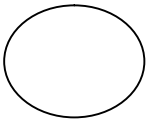
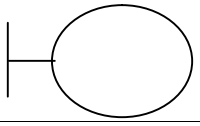
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points,* menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

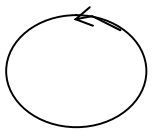
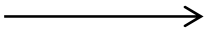
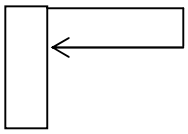
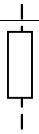
(Sumber : Windu Gata ; 2013 : 6)

4. *Sequence Diagram* (Diagram Rangkaian)

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
	*Entity Class,* 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.

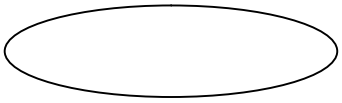
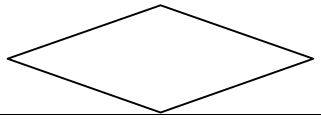
	*Control class,* suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive,* menggambarkan pengiriman pesan yang dikirim untuk dirinya 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 : Windu Gata ; 2013 : 7)

II.11. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) Al Fatta (2007:121) mengemukakan *Entity Relationship Diagram* adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis.Adapun simbolnya sebagai berikut :

Tabel II.5 Simbol-Simbol *Entity Relationship Diagram* (*ERD)*

No	Simbol	Keterangan Fungsi
1	Entitas 	Persegi panjang menyatakan himpunan entitas adalah orang, kajadian, atau berada dimana data akan dikumpulkan.
2	Atribut 	Atribut merupakan informasi yang diambil tentang sebuah entitas.
3	Relasi 	Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4	Link	Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya.

Sumber: (Ibnu Aqil; 2010: 6)

II.12. Kamus Data

Kamus data adalah suatu ensiklopedik dari informasi yang berkaitan dengan data perusahaan, atau dapat juga kita katakan bahwa kamus data adalah katalog atau *directory* yang berbasis komputer *(computer base catalog or*

directory) yang berbasis data perubahan (metadata). Yang berkenaan dengan tahapan penjelasan data ini adalah sistem kamus data (data *description language/*DDL). Sistem kamus data berbentuk perangkat lunak yang fungsinya adalah penciptaan dan pemeliharaan serta menyediakan kamus data agar dapat digunakan. Kamus data dapat berbentuk kertas ataupun arsip (*file*) komputer. (Ian Sommerville ; 2010 : 344)

II.13. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

II.13.1. Bentuk-bentuk Normalisasi

1. Bentuk normal tahap pertama (1” normal form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

p#	status	kota	b#	Qty
P1	20	Yogyakarta	B1	300
P2	20	Yogyakarta	B2	200

Tabel II.6 : Contoh bentuk normal pertama (1NF)

2. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. bentuk normal kedua jika dia berada pada bentuk normal kedua berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

p#	status	kota
P1	20	Yogyakarta
P2	20	Yogyakarta

P#	B#	qty
P1	B1	300
P1	B2	200
P1	B3	400

Tabel II.7 : Contoh bentuk normal kedua (2NF)

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

Pemasok kota

p#	kota
P1	Yogyakarta
P2	Yogyakarta

Kota_status

p#	kota
P1	Yogyakarta
P2	Yogyakarta

Tabel II.8 : Contoh bentuk normal ketiga (3NF)

4. **Boyce Code Normal Form (BCNF)**

   Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. **Bentuk Normal Tahap Keempat dan Kelima**

   Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan *multivalue* merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalue (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*). (Janner Simarmata ; 2010 : 76).