BAB II

TINJAUAN PUSTAKA

II.1. Definisi *Artificial Intelligenc*

Kecerdasan buatan (*Artificial Intelligence*) adalah Bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Menurut John McCarthy, 1956, AI untuk mengetahui dan memodelkan proses – proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia.

Penyakit asma telah dikenal sejak berabad-abad lalu dan sampai sekarang masih menjadi masalah kesehatan di masyarakat. Angka kejadian pada bayi dan anak lebih tinggi dibandingkan pada orang dewasa. Meskipun demikian, serangan asma untuk pertama kali tidak terlalu terjadi pada masa anak-anak. Serangan asma yang berat dapat menyebabkan kematian. Banyak faktor yang terlibat dalam terjadinya kematian karena asma. Akan tetapi, yang jelas 77 dari 90 kasus kematian karena asma dapat di cegah. Faktor-faktor utama penyebab kematian karena asma adalah ketidaktepatan diagnosis, penelitian beratnya asma oleh penderita maupun oleh dokter yang merawat kurang akurat, serta pengobatan yang kurang memadai. Oleh karena itu, ketepatan dalam diagnosis, penilaian beratnya asma, serta pemberian pengobatan yang tepat merupakan kunci pengobatan dalam serangan asma.(Muhammad Dahria ; 185)

II.2. Jaringan Saraf Tiruan

Jaringan saraf tiruan (*Artificial Nueral Network*) atau disingkat JST adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologis di dalam otak manusia, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. Model saraf ditunjukkan dengan kemampuannya dalam emulasi, analisis, prediksi dan asosiasi. Kemampuan yang dimiliki JST dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau input yang dimasukkan dan membuat prediksi tentang kemungkinan output yang akan muncul atau menyimpan karaktristik dari input yang disimpan kepadanya.

Valluru B.Rao dan Hayagriva V.Rao (1993) mendefenisi jaringan saraf sebagai sebuah kelompok pengolahan elemen dalam suatu kelompok yang khusus membuat perhitungan sendiri dan memberikan hasilnya kepada kelompok kedua atau berikutnya. Setiap sub kelompok menurut gilirannya harus membuat perhitungan sendiri dan memberikan hasilnya untuk subgrup atau kelompok yang belum melakukan perhitungan. Pada akhirnya sebuah kelompok dari satu atau beberapa pengolahan elemen tersebut menghasilkan keluaran (output) dari jaringan.

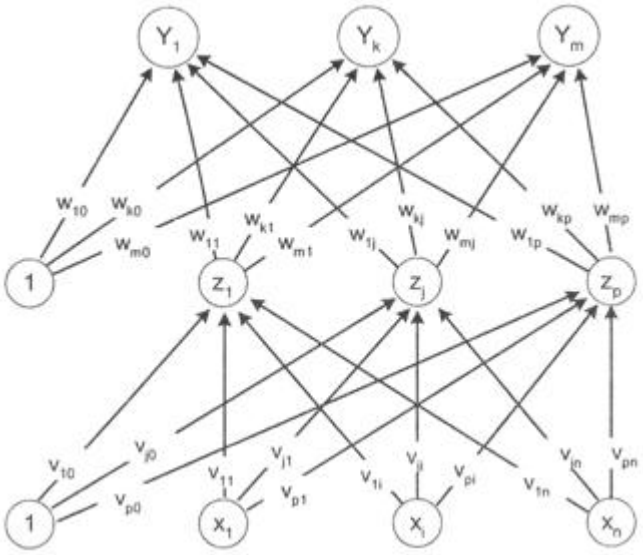
Setiap pengolahan elemen membuat perhitungan berdasarkan pada jumlah masukan (input). Sebuah kelompok pengolahan elemen disebut layer atau lapisan dalam jaringan. Lapisan pertama adalah input dan yang terakhir adalah output. Lapisan di antara lapisan input dan output disebut dengan lapisan tersembunyi

(hidden layer). Jaringan saraf tiruan merupakan suatu bentuk arsitektur yang terdistribusi paralel dengan sejumlah besar node dan hubungan antar node tersebut. Tiap titik hubungan dari satu node ke node yang lain mempunyai harga yang diasosiasikan dengan bobot. Setiap node memiliki suatu nilai yang diasosiasikan sebagai nilai aktivasi node.

Salah satu organisasi yang dikenal dan sering digunakan dalam paradigma jaringan saraf buatan adalah perambatan Galat Mundur (back-propagation). Sebelum dikenal adanya jaringan saraf perambatan Galat Mundur pada tahun 1950-1960-an,dikenal dua paradigma penting yang nantinya akan menjadi dasar dari saraf Perambatan Galat Mundur. (Relita Buaton ; 2014 : 134-135).

II.2.1. Arsitektur Backpropagation

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. Garnbar II.2 adalah arsitektur backpropagation dengan $_{n}$ buah masukan (ditambah sebuah bias), sebuah layar tersembunyi yang terdiri dari $_{p}$ unit (ditambah sebuah bias), serta $_{m}$ buah unit keluaran. $_{vii}$ merupakan bobot garis dari unit masukan $_{xi}$ ke rmit layar tersembunyi $_{z;}$ ($_{v;o}$ mempakan bobot garis yang menghuburngkan bias di unit masukan ke unit layar tersembunyi $_{z;}$).$_{w5}$ merupakan bobot dari unit layar tersembunyi $_{zi}$ ke unit keluaran $_{yk}$ ($_{wko}$ merupakan bobot dari bias di layar tersembunyi ke unit keluaran $_{zk}$.)



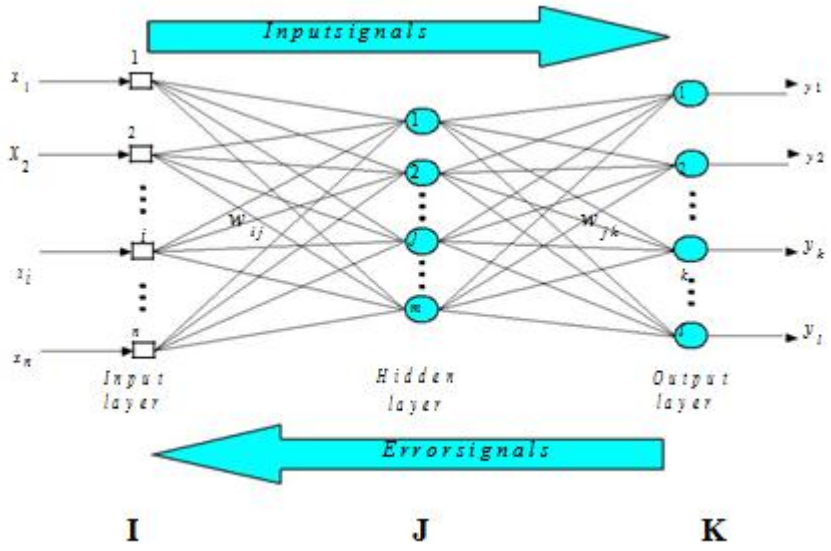
Gambar II.1. Arsitektur *Backpropagation*
(Sumber : Relita Buaton ; 2014 : 144-145)

II.3. Pengantar *Backpropagation*

Jaringan perambatan galat mundur (*backpropagation*) adalah salah satu algoritma yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit. Hal ini dimungkinkan karena pelatihan dengan menggunakan metode belajar terbimbing. Pada jaringan back propagation diberikan sepasang pola yang terdiri atas pola masukan dan pola yang diinginkan. Ketika suatu pola diberikan kepada jaringan, maka bobot-bobot diubah untuk memperkecil perbedaan pola keluaran dan pola yang diinginkan. Latihan ini dilakukan berulang-ulang sehingga semua pola yang dikeluarkan jaringan dapat memenuhi pola yang diinginkan.

Algoritma pelatihan jaringan saraf perambatan galat mundur terdiri atas dua langkah,yaitu perambatan maju dan perambatan mundur. Langkah perambatan mundur ini dilakukan pada jaringan untuk setiap pola yang

diberikan selama jaringan mengalami pelatihan. Jaringan perambatan galat mundur terdiri atas tiga lapisan atau lebih unit pengolah.



Gambar II.2. Arsitektur Backpropagation
(Sumber : Relita Buaton ; 2014 : 144,145)

Gambar II.1 menunjukkan jaringan perambatan galat mundur dengan tiga lapisan pengolah, bagian kiri sebagai masukan, bagian tengah disebut sebagai lapisan tersembunyi dan bagian kanan disebut lapisan keluaran. Ketiga lapisan ini terhubung secara penuh. Perambatan maju dimulai dengan memberikan pola masukan ke lapisan masukan. Pola masukan ini merupakan nilai aktivasi unit-unit masukan. Dengan melakukan perambatan maju dihitung nilai aktivasi pada unit-unit di lapisan berikutnya. Pada setiap lapisan,tiap unit pengolah melakukan penjumlahan berbobot dan menerapkan fungsi sigmoid untuk menghitung keluarannya. (Relita Buaton ; 2014 : 144-145).

II.3.1. Rumus *Backpropagation*

1. Keluaran Hiden layer dengan menggunakan rumus

$$Yj(p) = Sigmoid\left[\sum_{i=1}^{N} xi(p)\, x\, wij(p) - \theta j\right]$$

Dimana

Yj=keluaran unit j

Sigmoid=fungsi aktivasi

Xi=input dari unit i

Wij=bobot dari unit i ke j

θj =batas ambang unit j

P=iterasi

2. Keluaran Output layer dengan menggunakan rumus

$$Yk = sigmoid\left[\sum_{j=1}^{M} Xjk(p)\, x\, Wjk(p) - \theta k\right]$$

Dimana

Yk=keluaran unit k

Sigmoid=fungsi aktivasi

Xjk=input dari unit j

Wjk=bobot dari unit j ke k

θk =batas ambang unit k

P=iterasi

3. Menentukan Erorr dengan menggunakan rumus

$$ek(p) = Yd,k(p) - Yk(p)$$

Dimana

ek=error unit k(output layer)

Ydk=output yang diharapkan pada unit k

Yk=output actual pada unit k

P=iterasi

II.3.2. Langkah-Langkah Penyelesaian Backpropagation

1. Inisialisasi

   Tentukan input, output yang diharapkan, bobot input, bobot hidden, treshold hidden, threshold output, training rate,

2. Hitung keluaran hidden

$$Yj(p) = Sigmoid\left[\sum_{i=1}^{N} xi(p)\, x\, wij(p) - \theta j\right]$$

   Gunakan fungsi aktivasi Ysigoid untuk menentukan keluaran hidden dan output layer dengan rumus

$$Y^{Sigmoid} = \frac{1}{1 + e^{-X}}$$

3. Hitung keluaran output

$$Yk = sigmoid\left[\sum_{j=1}^{M} Xjk(p)\, x\, Wjk(p) - \theta k\right]$$

4. Hitung error dengan rumus $ek(p) = Yd,k(p) - Yk(p)$

5. Update bobot hidden(Wjk) a. Hitung gradien

   error

$$\delta k\,(p) = Yk(p)\; x\; [1 - Yk(p)]\; x\; ek(p)$$

   b. Update bobot

$$Wjk(p+1) = Wjk(p) + \Delta Wjk(p)$$

$$\Delta Wjk(p) = \alpha\; x\; Yj(p)\; x\; \delta k(p)$$

6. Update bobot input(Wij)

   a. Hitung gradient error

$$\delta j(p) = Yj(p)\; x\; [1 - Yj(p)]\; x \sum_{k=1}^{l} \delta k\,(p)\; x\; Wjk\,(p)$$

   b. Update bobot

$$Wij(p+1) = Wij(p) + \Delta Wij(p)$$

$$\Delta Wij(p) = \alpha\; x\; Xi(p)\; x\; \delta j(p)$$

II.4. Pengertian Sistem

Secara leksikal, sistem berarti susunan yang teratur dari pandangan, teori, asas dan sebagainya. Dengan kata lain, sistem adalah suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antara bagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interpenden satu sama lain. Selain itu dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan-perubahan yang terus menerus perlu dikembangkan dan dikendalikan. Definisi tersebut menunjukkan bahwa sistem sebagai gugus dan elemen-elemen

yang saling berinteraksi secara teratur dalam rangka mencapai tujuan atau subtujuan (Marimin, 2008 : 1).

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari :

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

   Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkung luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

   Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem

   Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

   Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

   Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya (Sulindawati, 2010 : 375).

II.5. Pengertian Java

Bahasa pemrograman Java merupakan karya SunsMicrosystemInc. Rilis resmi level *beta* dilakukan pada November 1995. Dua bulan berikutnya Netscape menjadi perusahaan pertama yang memperoleh lisensi bahasa Java dari Sun. Java adalah bahasa pemrograman berorientasi objek yang berukuran kecil, sederhana dan aman, diinterpretasi atau dioptimalisasi secara dinamis, ber-*bytecode*, arsitektur yang netral, mempunyai *garbage-collector, multithreading*, mempunyai

mekanisme penanganan pengecualian, berbasis tipe untuk penulisan program mudah diperluas dinamis serta telah diperuntukkan sistem tersebar.

Pada awal pembuatannya, Java dinamakan Oak, kemudian nama Oak dinilai kurang menjual sehingga pada Januari 1995 nama Oak diubah menjadi Java. Sebagai bahasa yang bersifat terbuka, Java didukung oleh banyak *programmer* dari seluruh dunia yang memberikan kontribusinya untuk mengembangkan bahasa Java (Didik Dwi Prasetyo ; 2010 : 1).

Pada pengembangan *enterpriseapplications*, kita menggunakan sejumlah besar paket. Pada *consumerelectronicproduct*, hanya sejumlah kecil bagian bahasa yang digunakan. Masing-masing edisi berisi *Java 2 Software Development KIT* (SDK) untuk mengembangkan aplikasi dan *Java 2 Runtime Environment* (JRE) untuk menjalankan aplikasi.

a. *Standard Edition*(J2SE)

   The Java 2 Platform, Standard Edition(J2SE) menyediakan lingkungan pengembangan yang kaya fitur, stabil, aman dan *cross-platform*. Edisi ini mendukung konektivitas basis data, rancangan antarmuka pemakai, masukan/keluaran dan pemrograman jaringan dan termasuk sebagai paket-paket dasar bahasa Java.

b. *EnterpriseEdition*(J2EE)

   The Java 2, EnterpriseEdition(J2EE) menyediakan kakas untuk membangun dan menjalankan *multitierenterpriseapplications*. J2EE berisi paket-paket di J2SE ditambah paket-paket untuk mendukung

pengembangan *EnterpriseJavaBeans, Java Servlets, JavaServer, Pages, XML,* dan kendali transaksi yang fleksibel.

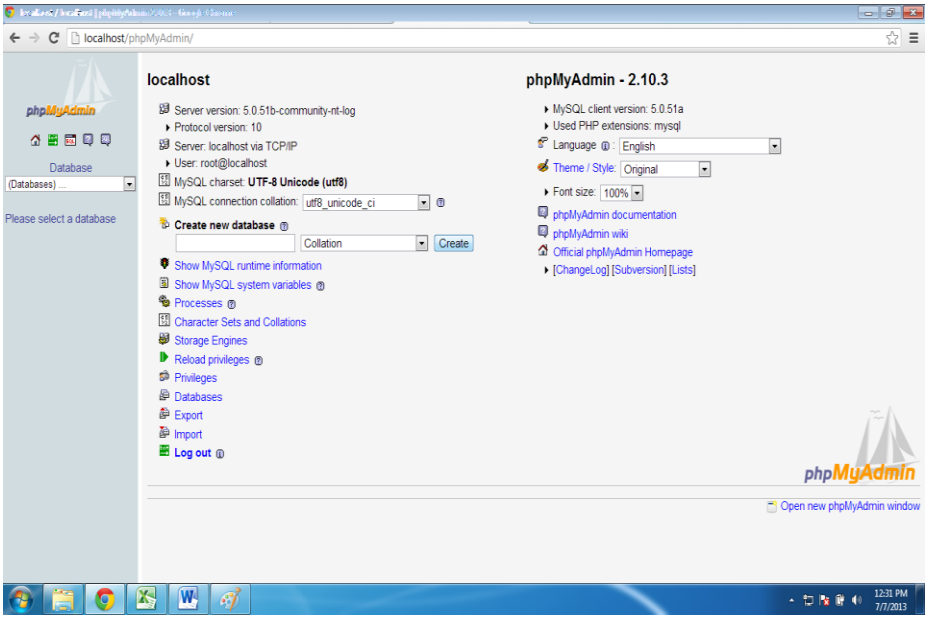
c. *Micro Edition*(J2ME)

The Java 2, Micro Edition (J2ME) untuk beragam *consumer electronic product,* seperti *pager, smartcard, cellphone, handheld* PDA, dan *settopbox.* J2ME sembari menyediakan bahasa Java yang sama, unggul dalam portabilitas kemampuan dijalankan dimana pun dan *safe network delivery* seperti J2SE dan J2EE.(Didik Dwi Prasetyo ; 2010 : 3).

II.6. Pengertian MySQL

Mysql pertama kali dirintis oleh seorang programmer database bernama Michael Widenius, yang dapat anda hubungi di emailnya monty@analytikerna. Mysql *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer di antara *database* yang lain.

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*) (Wahana Komputer; 2010 : 5).

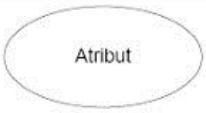


Gambar II.3. Tampilan MySQL
(Sumber : Wahana Komputer, 2010 : 5)

II.7. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram atau ERD adalah alat pemodelan berbentuk grafis, yang popular untuk desain database. Mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antarentitas. Proses memungkinkan analis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Yuniar Supardi, 2010 : 448).

Tabel II.1. Simbol ERD

Notasi	Keterangan
	**Entitas**, adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	**Relasi**, menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	**Atribut**, berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

(Sumber : Yuniar Supardi, 2010 : 448)

II.8. 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.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang tekah ditemukan.

II.8.1.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.

2. **Bentuk normal tahap kedua (2^{nd} normal form)**

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

3. **Bentuk normal tahap ketiga (3^{rd} normal form)**

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

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*) (Yuniar Supardi, 2010 : 448).

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

Menurut Windu Gata (2013 : 4) 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.

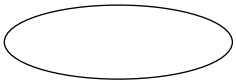
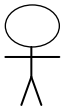
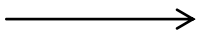
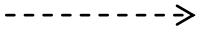
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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, yaitu :

Tabel II.2. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan 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.

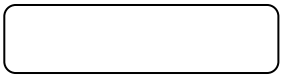
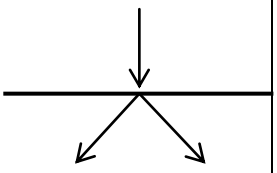
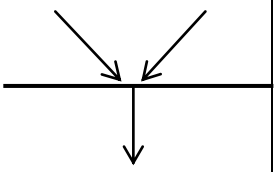
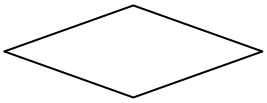
<--------	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata, 2013 : 4)

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*, yaitu :

Tabel II.3. Simbol *Activity Diagram*

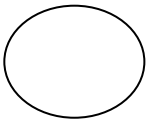
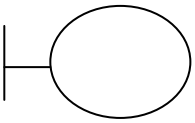
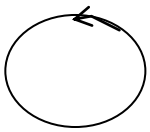
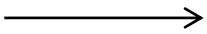
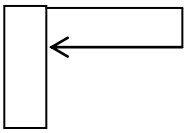
Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan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

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*, yaitu :

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)

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.

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 : Windu Gata, 2013 : 9)