

BAB II

TINJAUAN PUSTAKA

II.1. Kecerdasan Buatan

Menurut Muhammad Dahria (2008), Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian ilmu computer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia bahkan bias lebih baik yang dilakukan pada manusia.

Muhammad Dahria (2008) juga mengatakan, AI : untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), dan moral yang baik.

Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga memiliki akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk melakukan penalaran dengan baik, manusia dengan segudang pengalaman tidak akan dapat menyelesaikan masalah dengan baik.

II.2. Data Mining

Menurut Joanna Ardhyanti Mita Nugraha dan Yupie Kusumawati, *Data mining* merupakan proses pencarian pola-pola yang menarik dan tersembunyi (*hidden pattern*) dari suatu kumpulan data yang berukuran besar yang tersimpan dalam suatu basis data, *data warehouse*, atau tempat penyimpanan data lainnya

Data mining (DM) adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara *manual*. Patut diingat bahwa kata *mining* sendiri berarti usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar. Karena itu DM sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, statistik dan basis data. Beberapa teknik yang sering disebut-sebut dalam *literatur* DM antara lain : *clustering*, *classification*, *association rule mining*, *neural network*, dan *genetic algorithm*. (Lindawati:2008).

Data mining bukan hanya pelengkap saja dalam suatu database, melainkan mempunyai fungsi yang penting untuk membantu penggunanya mendapatkan informasi yang berguna serta meningkatkan pengetahuan bagi sang pengguna itu sendiri dan dapat nantinya berguna untuk orang banyak. Pada dasarnya, *data mining* mempunyai empat fungsi dasar yaitu (Sando Romario, 2003) :

1. Fungsi Klasifikasi (*classification*)

Data mining dapat digunakan untuk mengelompokkan data-data yang jumlahnya besar menjadi data-data yang lebih kecil.

2. Fungsi Segmentasi (*Segmentation*)

Disini data mining juga digunakan untuk melakukan segmentasi (pembagian) terhadap data berdasarkan karakteristik tertentu.

3. Fungsi Asosiasi (*Association*)

Disini data mining juga digunakan untuk mencari hubungan antara karakteristik tertentu .

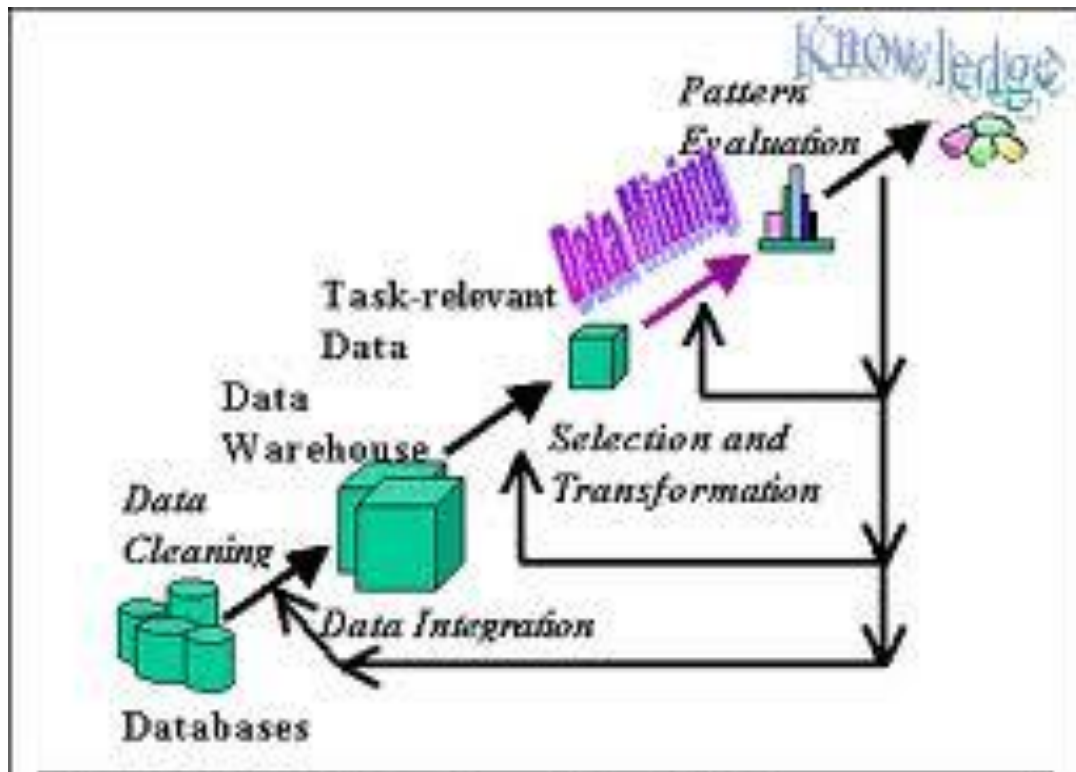
4. Fungsi pengurutan (*Sequencing*)

Pada Fungsi ini, data mining digunakan untuk mengidentifikasi perubahan pola yang telah terjadi dalam jangka waktu yang tertentu.

II.2.1. Tahapan Data Mining

Menurut Lindawati, Data Mining dapat dibagi menjadi beberapa tahap. Tahap-tahap tersebut bersifat *interaktif* di mana pemakai terlibat langsung atau dengan perantaraan *knowledge base*.

Data yang ada, tidak dapat langsung diolah dengan menggunakan sistem *Data Mining*. Data tersebut harus dipersiapkan terlebih dahulu agar hasil yang diperoleh dapat lebih maksimal, dan waktu komputasinya lebih minimal. Proses persiapan data ini sendiri dapat mencaapai 60% dari keseluruhan proses dalam *Data Mining*. Tahap-tahap ini diilustrasikan di Gambar II.1 berikut ini:



Gambar II.1 Tahap-Tahap Data Mining
(Sumber: Romario, 2013)

Menurut Sando Romarios (2013) untuk lebih mengetahui mengenai tahapan-tahapan proses pada *knowledge in databases*, berikut penjelasan mengenai tahapan-tahapan tersebut menurut :

1. *Database*

Koleksi data yang saling berhubungan untuk dipergunakan secara bersama kemudian dirancang untuk memenuhi kebutuhan informasi organisasi

2. *Data Cleaning*

Pada umumnya, data yang diperoleh, baik dari database suatu perusahaan maupun eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau hanya sekedar salah ketik. Data-data

yang tidak relevan itu juga lebih baik dibuang karena keberadaanya bisa mengurangi mutu atau akurasi dari hasil data mining nantinya. Pembersihan data juga akan mempengaruhi performansi dari sistem data mining karena data yang akan ditangani akan berkurang jumlah dan kompleksitasnya.

3. *Data integration*

Integrasi data dilakukan pada atribut-attribut yang mengidentifikasi *entitas-entitas* yang unik seperti atribut nama, nomor pegawai, tempat lahir, agama dan lain sebagainya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil data yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya.

4. *Task relevan data*

Setelah semua sumber data digabung atau diintegrasikan menjadi satu keseluruhan *database*, maka tahap selanjutnya adalah melakukan *task relevant data*. Pada tahap ini melakukan relevansi atribut dari data yang *relevant* atau yang sesuai dengan target atau *output* yang akan dihasilkan.

5. *Data transformation*

Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam *database*.

6. *Data mining*

Data mining adalah proses mengeksplorasi dan menganalisa data dalam

jumlah yang besar yang bertujuan untuk menemukan suatu pola atau informasi yang menarik dari data yang tersimpan dalam jumlah yang besar dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan. Tahap ini merupakan inti dari tahapan KDD yang dilakukan untuk menganalisis data yang telah dibersihkan.

7. *Pattern evaluation*

Dalam tahap ini, merupakan hasil dari teknik data mining berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai dengan hipotesa, ada beberapa alternatif yang dapat diambil seperti menjadikannya umpan balik untuk memperbaiki *data mining* lain yang lebih sesuai, atau menerima hasilnya sebagai suatu hasil yang diluar dugaan yang mungkin bermanfaat.

8. *Knowledge*

Tahap terakhir dari proses *data mining* adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisa yang didapat. Ada kalanya hal ini harus melibatkan orang-orang yang tidak paham mengenai *data mining*. Karenanya presentasi hasil *data mining* dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses *data mining*. Dalam presentasi ini, *visualisasi* juga bisa membantu mengkomunikasikan hasil data mining.

II.2.2 Pengelompokan *Data Mining*

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat di lakukan, yaitu (Larose, 2006):

1. Deskripsi

Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.

4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan suatu dengan yang lainnya dan memiliki ketidakmiripan dengan *record* dalam kluster lain.

6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam suatu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

II.2.3 Clustering

Menurut Joanna Ardhyanti Mita Nugraha dan Yupie Kusumawati menyatakan *Clustering* adalah mengelompokkan item data ke dalam sejumlah kecil grup sedemikian sehingga masing-masing grup mempunyai sesuatu persamaan yang esensial. Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang

memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan suatu dengan yang lainnya dan memiliki ketidakmiripan dengan *record* dalam kluster lain.

Menurut Lindawati, *Data Mining* dengan teknik *clustering*, berbeda dengan teknik *association rule mining* dan *classification* dimana kelas data telah ditentukan sebelumnya. *Clustering* melakukan pengelompokan data tanpa berdasarkan kelas data tertentu. Bahkan *clustering* dapat dipakai untuk memberikan label pada kelas data yang belum diketahui itu. Karena itu *clustering* sering digolongkan sebagai metode *unsupervised learning*. Prinsip dari *clustering* adalah memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas atau *cluster*. *Clustering* dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai ruang multidimensi.

Pada dasarnya *clustering* merupakan suatu metode untuk mencari dan mengelompokkan data yang memiliki kemiripan karakteristik (*similarity*) antara satu data dengan data yang lain. *Clustering* merupakan salah satu metode *data mining* yang bersifat tanpa arahan (*unsupervised*), maksudnya metode ini diterapkan tanpa adanya latihan (*training*) dan tanpa ada guru (*teacher*) serta tidak memerlukan target output. Dalam data mining ada dua jenis metode *clustering* yang digunakan dalam pengelompokan data, yaitu *hierarchical clustering* dan *non-hierarchical clustering* (Santosa, 2007).

Clustering adalah sebuah kegiatan dalam mengelompokkan berbagai macam rekaman data, observasi, atau melakukan pendekatan data dari sebuah objek dengan objek lain yang memiliki kesamaan tertentu. *Cluster* merupakan

kumpulan baris data yang memiliki kesamaan dan ketidaksamaan untuk merekam kedalam *cluster* lain. (Johan Oscar Ong, 2013).

K-means clustering merupakan salah satu metode data clustering non-hirarki yang mengelompokkan data dalam bentuk satu atau lebih cluster/kelompok. Data-data yang memiliki karakteristik yang sama dikelompokkan dalam satu *cluster*/kelompok dan data yang memiliki karakteristik yang berbeda dikelompokkan dengan *cluster*/kelompok yang lain sehingga data yang berada dalam satu *cluster*/kelompok memiliki tingkat variasi yang kecil (Agusta, 2007).

Menurut D.T Larose (2007), langkah-langkah melakukan clustering dengan metode *K-Means* adalah sebagai berikut:

1. Tentukan Jumlah *Cluster* Yang Ingin Dibentuk Dari Sebuah *Data Set*
2. Tentukan lokasi tengah atau *centroid cluster* secara acak
3. Untuk setiap baris data, temukan *cluster* yang paling dekat dengan pusat *cluster*.
4. Untuk setiap *cluster*, temukan *centroid cluster* dan perbaharui lokasi masing-masing pusat *cluster* untuk mendapatkan nilai *centroid* yang baru
5. Ulangi langkah ke-3 sampai langkah ke-5 sampai ditemui *cluster* yang konvergen sebagai akhir dari *clustering*.

Rumus untuk menghitung jarak diantara dua titik dengan *Euclidean distance space* menggunakan rumus:

$$d_{Euclidean}(x,y) = \sqrt{\sum(x_i - y_i)} \dots\dots\dots (1)$$

Dimana:

x dan y : representasi nilai atribut dari dua *record*.

Sedangkan untuk membangkitkan *centroid baru* dapat menggunakan rumus pembangkitan *centroid* berikut ini:

$$C = \frac{\sum m}{n} \dots\dots\dots (2)$$

Dimana:

C : *Centroid Data*

m : Anggota Data Yang Termasuk
Kedalam *Centroid Tertentu*

n : Jumlah Data Yang Menjadi
Anggota *Centroid Tertentu*

II.3. *Unified Modelling Language (UML)*

Menurut Sri Dharwiyanti dan Romi Satria Wahono, *Unified Modelling Language (UML)* adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa

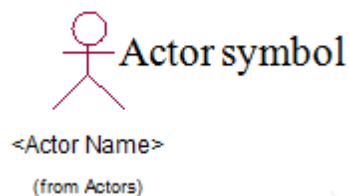
berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

II.3.1. Notasi dalam UML

Menurut Jamilah, 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. Berikut notasi-notasi pada UML:

1. Aktor

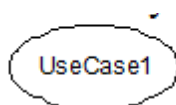
Aktor menggambarkan segala pengguna software aplikasi (user). Actor memberikan suatu gambaran jelas tentang apa yang harus dikerjakan software aplikasi. Sebagai contoh sebuah actor dapat memberikan input kedalam dan menerima informasi dari software aplikasi, perlu dicatat bahwa sebuah actor berinteraksi dengan use case, tetapi tidak memiliki kontrol atas use case. Sebuah actor mungkin seorang manusia, satu device, hardware atau sistem informasi lainnya.



Gambar II.2 : Notasi Aktor
(Sumber : Haviluddin : 2011)

2. Use Case

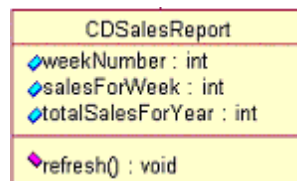
Use case menjelaskan urutan kegiatan yang dilakukan actor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, namun use case hanya menjelaskan apa yang dilakukan oleh actor dan sistem bukan bagaimana actor dan sistem melakukan kegiatan tersebut.



Gambar II.3: Notasi Use Case
(Sumber : Haviluddin : 2011)

3. Class

Class merupakan pembentuk utama dari sistem berorientasi obyek, karena class menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama. Class digunakan untuk mengimplementasikan interface. Class digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun. Class bisa merepresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata. Notasi class berbentuk persegi panjang berisi 3 bagian: persegi panjang paling atas untuk nama class, persegi panjang paling bawah untuk operasi, dan persegi panjang ditengah untuk atribut. Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas merepresentasikan informasi yang tersimpan didalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh obyek dan menggunakan kata kerja.



Gambar II.4: Notasi Class
(Sumber : Haviluddin : 2011)

4. Interface

Interface merupakan kumpulan operasi tanpa implementasi dari suatu class. Implementasi operasi dalam interface dijabarkan oleh operasi didalam class. Oleh karena itu keberadaan interface selalu disertai oleh class yang mengimplementasikan operasinya. Interface ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam obyek.



Gambar II.5: Notasi Interface
(Sumber : Haviluddin : 2011)

5. Interaction

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya interaction ini dilengkapi juga dengan teks bernama operation signature yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.



Gambar II.6: Notasi Interaction

(Sumber : Haviluddin : 2011)

6. Note

Note digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. Note ini bias disertakan ke semua elemen notasi yang lain.

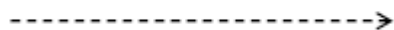


Gambar II.7: Notasi Note

(Sumber : Haviluddin : 2011)

7. Depedency

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada dibagian tanpa tanda panah.



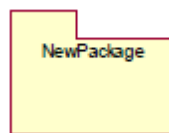
Gambar II.8: Notasi Dependence

(Sumber : Haviluddin : 2011)

8. Package

Package adalah mekanisme pengelompokkan yang digunakan untuk menandakan pengelompokkan elemen-elemen model. Sebuah paket dapat

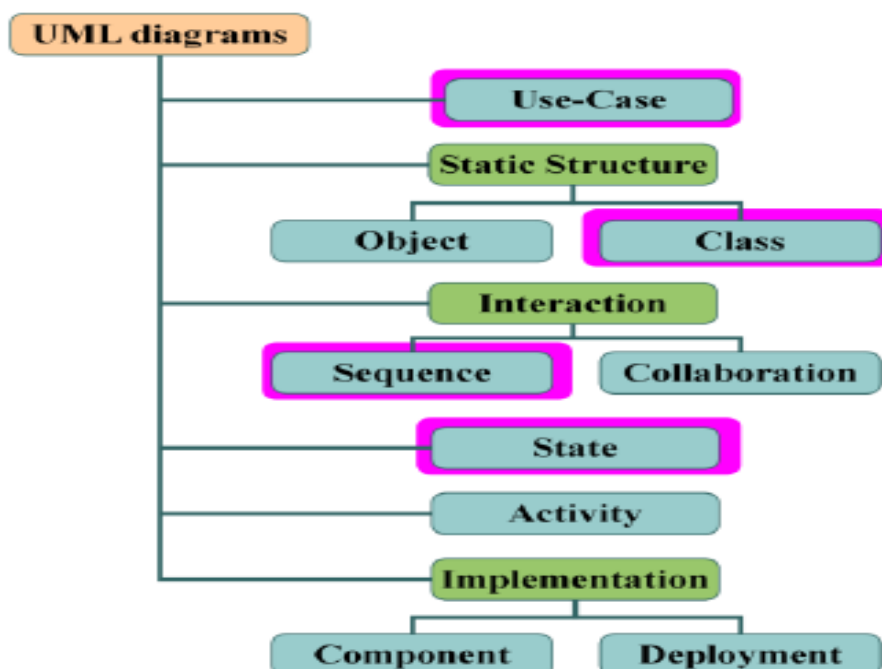
mengandung beberapa paket lain didalamnya. Paket digunakan untuk memudahkan mengorganisasikan elemen-elemen model.



Gambar II.9: Notasi Package
(Sumber : Havaluddin : 2011)

II.3.2. Diagram dalam UML

UML sendiri terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model.



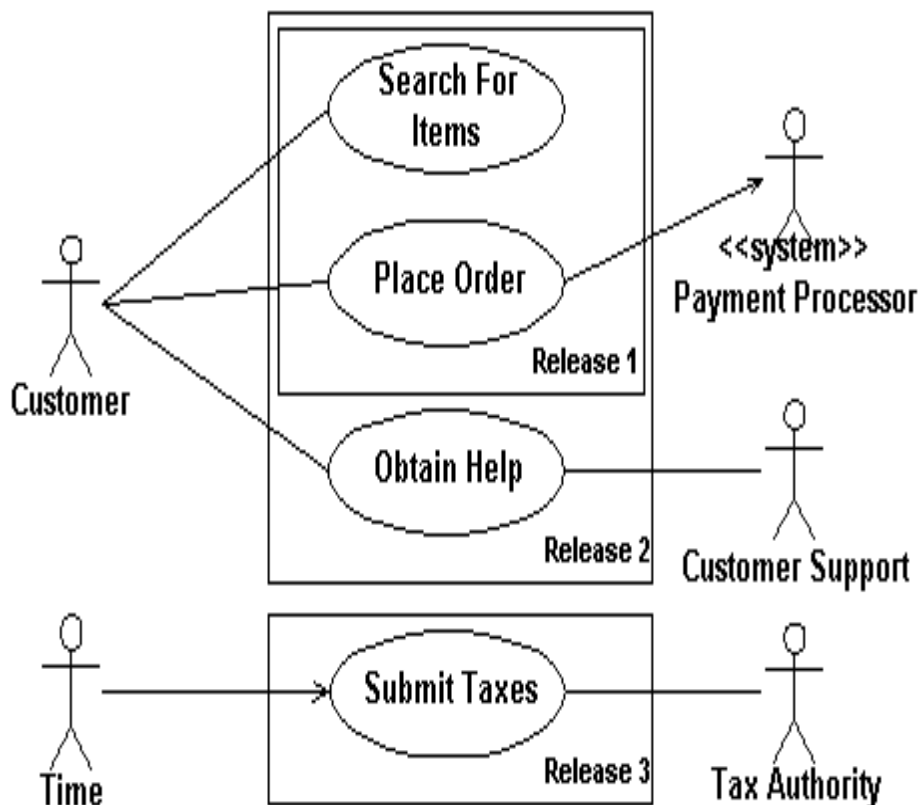
Gambar II.10: Diagram UML
(Sumber: Havaluddin,2011)

Adapun penjelasan dari diagram diatas yaitu (Haviluddin, 2011):

1. Use Case Diagram

Menurut Haviluddin, *Use Case Diagram* yaitu Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. *Use Case* memiliki dua istilah

1. *System use case*; interaksi dengan sistem.
2. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata



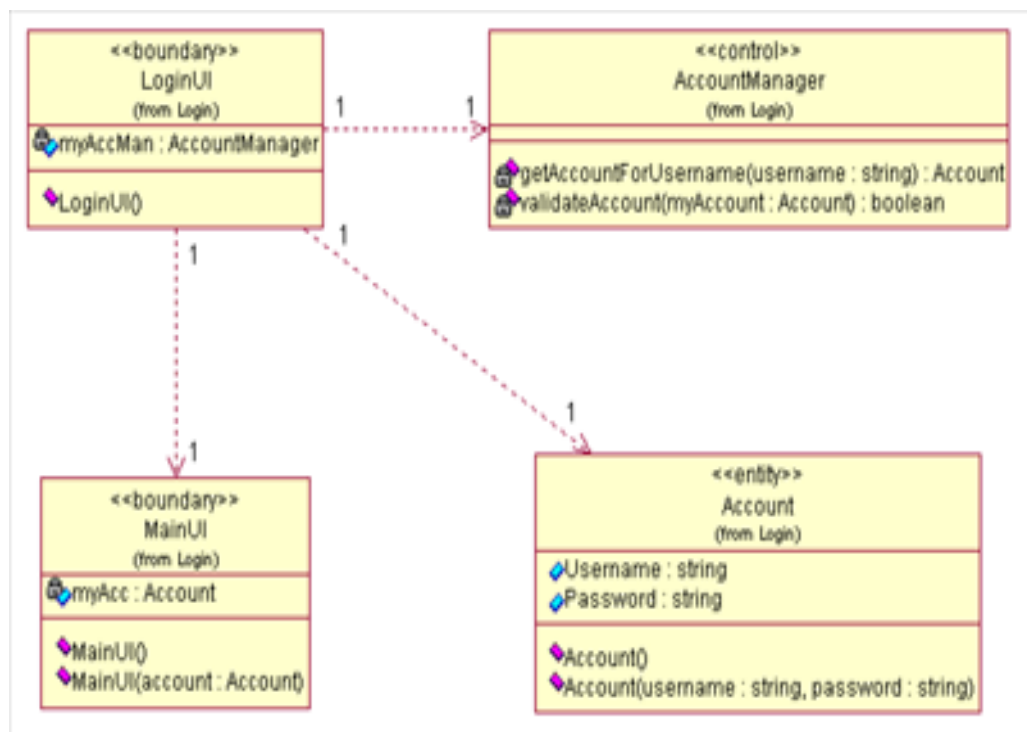
Gambar II.11: Use Case Diagram

(Sumber: Haviluddin, 2011)

2. Class Diagram

Menurut Haviluddin, *Class Diagram* adalah menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok :

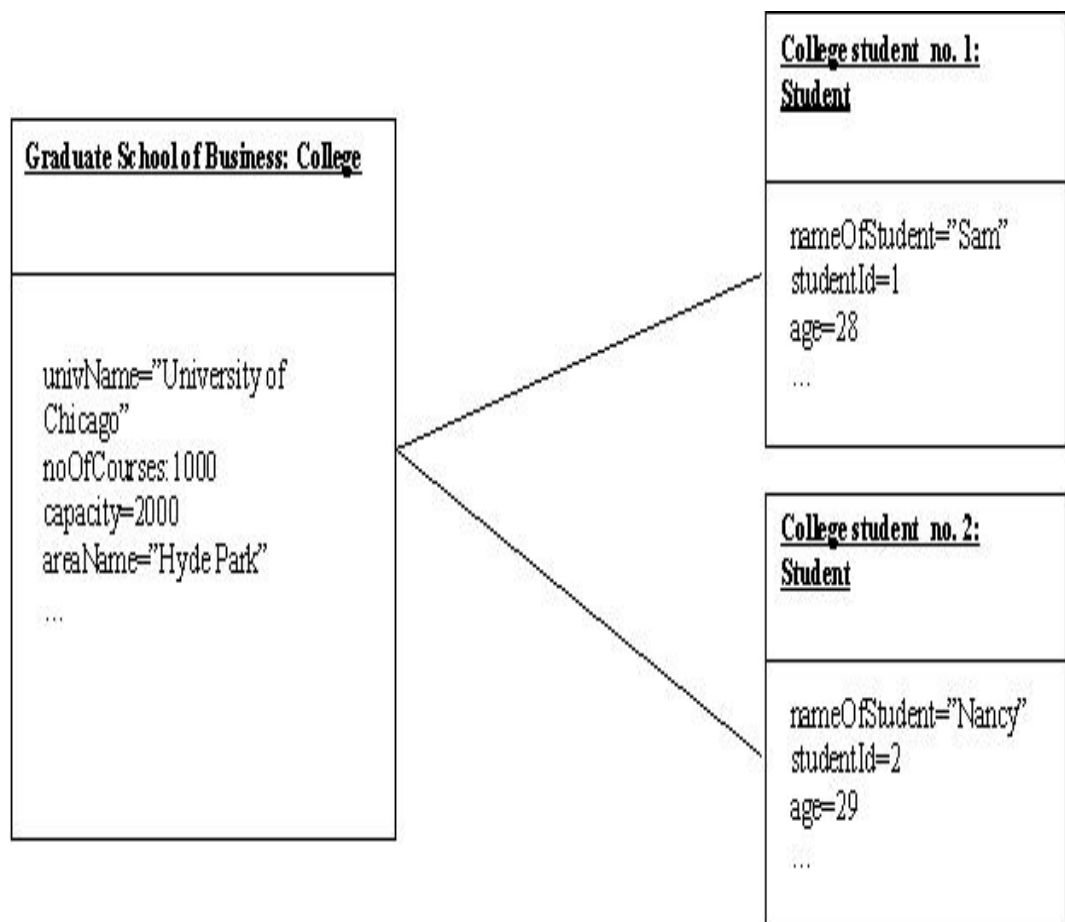
1. Nama (dan *stereotype*)
2. Atribut
3. Metoda



Gambar II.12: Class Diagram
(Sumber: Haviluddin, 2011)

3. Object Diagram

Menurut Havaluddin, *Object* diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.

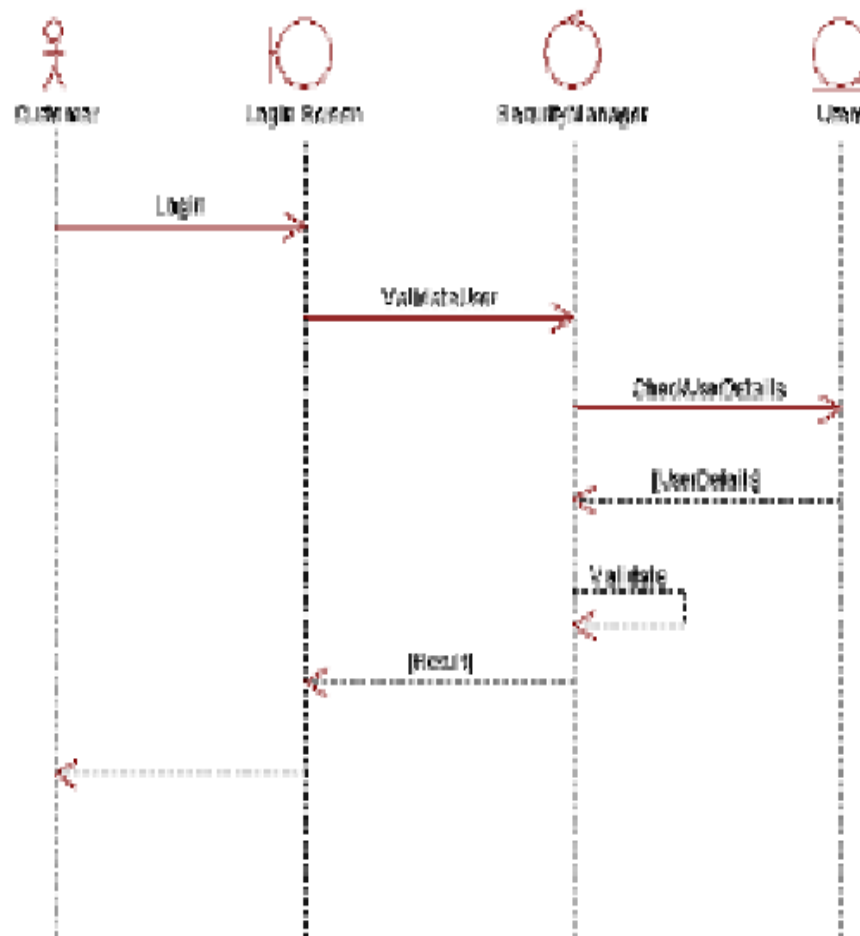


Gambar II.13: Object Diagram
(Sumber: Havaluddin, 2011)

4. Sequence Diagram

Menurut Havaluddin, *Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah

gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.



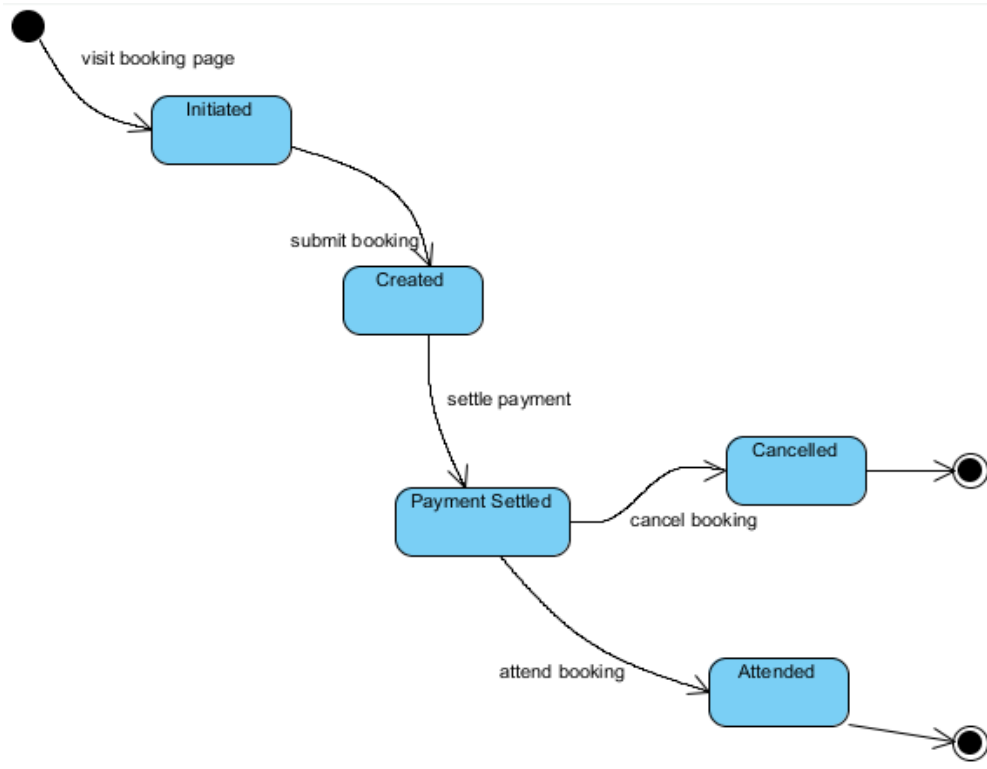
Gambar II.14: *Sequence Diagram*

(Sumber: Havaluddin, 2011)

5. *Statechart Diagram*

Menurut Havaluddin, *Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram*

menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

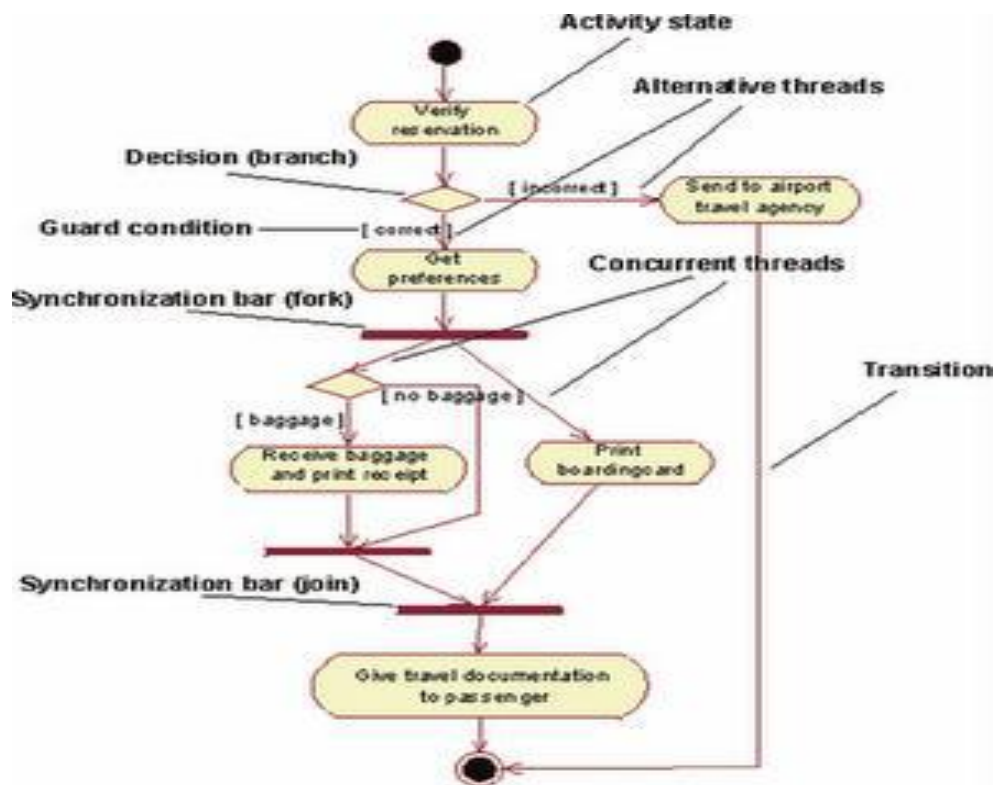


Gambar II.15: Statechart Diagram

(Sumber: Havaluddin, 2011)

6. Activity Diagram

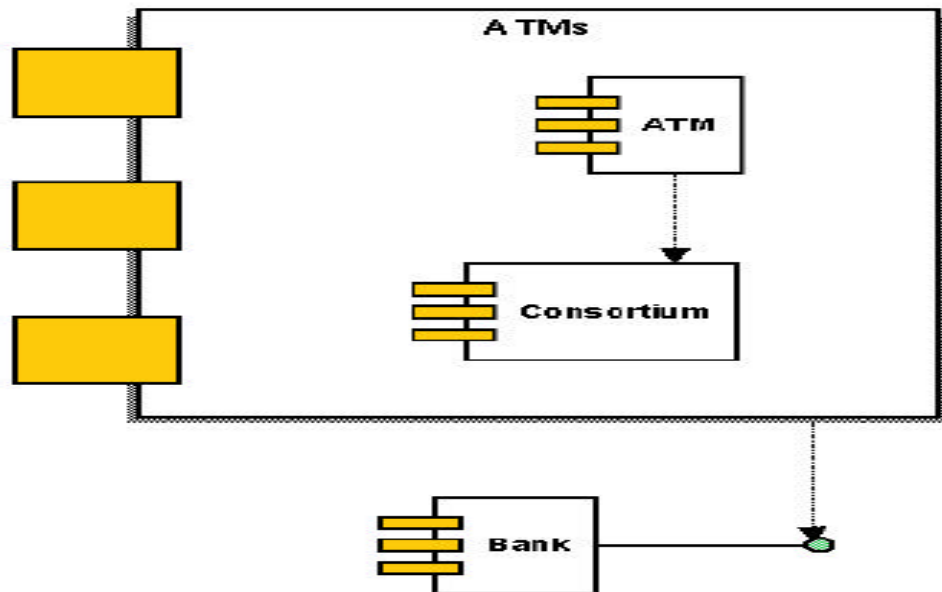
Menurut Havaluddin, *Activity Diagram* Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas



Gambar II.16: Activity Diagram
(Sumber: Havaluddin, 2011)

7. Component Diagram

Menurut Havaluddin, *Component* diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.

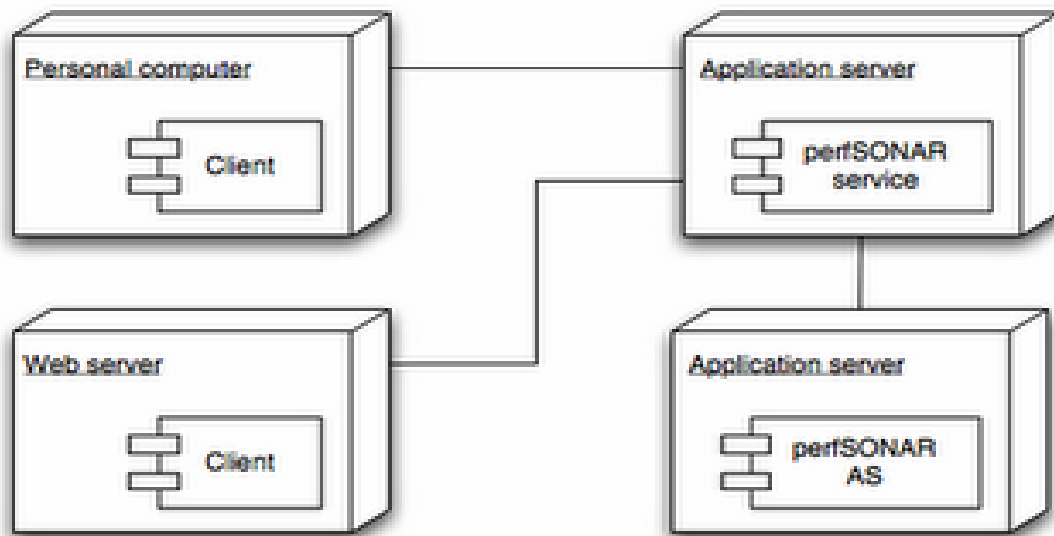


Gambar II.17: Component Diagram

(Sumber: Haviluddin, 2011)

8. Deployment Diagram

Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.



Gambar II.18: *Deployment Diagram*
(Sumber: Haviluddin, 2011)

II.4. Pengertian Basis Data

Menurut Date (200, p.10) basis data merupakan kumpulan data tetap (data yang hampir tidak mengalami perubahan), yang digunakan oleh sistem aplikasi di beberapa perusahaan.

Menurut Connely dan Begg (2002, p.14) basis data adalah kumpulan data yang terkumpul secara logis, dan merupakan deskripsi dari data tersebut, dirancang untuk mempermudah proses pencarian informasi yang dibutuhkan oleh perusahaan.

Dari beberapa defenisi tentang basis data diatas, dapat disimpulkan bahwa basis data merupakan kumpulan data yang terhubung secara logis dan digunakan pada system aplikasi perusahaan, yang dapat mempermudah proses pencarian informasi.

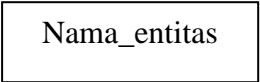
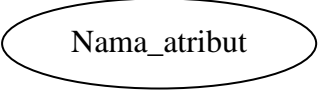
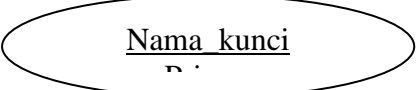
II.4.1. Entity Relationship Diagram (ERD)

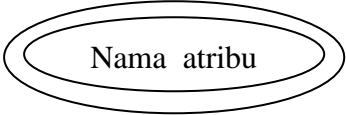
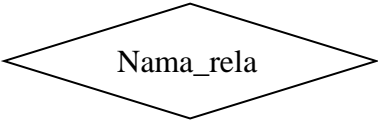

Entity Relationship Diagram (ERD) adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek kedalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien (Gallaleo I, dkk, 2014 : 12).

Demikian pula untuk membantu menggambarkan relasi secara lengkap terdapat juga beberapa relasi dalam hubungan atribut yang ada dalam satu atau dua file.

Didalam sebuah ERD, terdapat simbol-simbol. Adapun simbol-simbol ERD (*Entity Relationship Diagram*) ditunjukkan pada tabel II.1.

Tabel II.1. Simbol-simbol ERD (*Entity Relationship Diagram*)

Simbol	Deskripsi
Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data
Atribut 	<i>Field</i> atau kolom data yang perlu disimpan dalam suatu entitas
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan;

	biasanya berupa id
Atribut multivalued/ <i>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antarentitas; relasi biasanya diawali dengan kata kerja
Asosiasi/ <i>Association</i> 	Penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian

Sumber: (Rossa A.S-M. Shalahuddin;2011:50-51)

II.4.2. Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan atau mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomalies*) yang terjadi adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan (Edy Sutanta ; 2011 : 174)

Proses normalisasi menghasilkan relasi yang optimal, yaitu (Edy Sutanta ; 2011 : 175).

1. Memiliki struktur *record* yang konsisten secara logic.
2. Memiliki struktur *record* yang mudah untuk dimengerti.
3. Struktunya yang memiliki *record* yang sederhana dalam suatu pemeliharaan.
4. Struktur *record* yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna.
5. Minimalisasi kerangkapan data guna meningkatkan kinerja suatu sistem.

Tahapan normalisasi terdiri dari beberapa bentuk yaitu sebagai berikut:

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

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

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

Defenisi 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 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk Normal Ketiga (3NF/*Third Normal Form*)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung pada kunci utama. Secara defenisi, 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. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama.

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

Bentuk normal *Boyce Code* (BCNF) adalah versi 3NF yang lebih teliti dan berhubungan dengan tabel relasional yang mempunyai (a) banyak kunci kandidat, (b) kunci kandidat gabungan, dan (c) kunci kandidat yang saling tumpang tindih.

BCNF didasarkan pada konsep penentu. Sebuah kolom penentu adalah kolom di mana kolom-kolom lain sepenuhnya tergantung secara fungsional. Sebuah tabel relasional berada pada BCNF jika dan hanya jika setiap penentu adalah kunci kandidat.

II.5. SQL Server 2008 Express

Microsoft SQL Server merupakan produk *Relational Database Management System* (RDBMS) yang dibuat oleh Microsoft. Orang sering menyebutnya dengan SQL Server saja. Microsoft SQL Server juga mendukung SQL sebagai bahasa untuk memproses query ke dalam database. Microsoft SQL

Server banyak digunakan pada dunia bisnis, pendidikan atau juga pemerintahan sebagai solusi database atau penyimpanan data. Pada tahun 2008 Microsoft mengeluarkan SQL Server 2008 yang merupakan versi yang banyak digunakan.

II.6 Visual Basic 2010

Microsoft Visual Studio 2010 merupakan sebuah IDE (*Integrated Development Environment*) yang dikembangkan oleh Microsoft. IDE ini mencakup semua bahasa pemrograman berbasis *.NET framework* yang dikembangkan oleh Microsoft. Keunggulan Microsoft Visual Studio 2010 ini antara lain adalah *support* untuk Windows 8, editor baru dengan WPF (*Windows Presentation Foundation*) dan banyak peningkatan fitur lainnya.