

BAB II

TINJAUAN PUSTAKA

II.1. *Data Mining*

II.1.1. Pengertian *Data Mining*

Nama *data mining* sebenarnya mulai dikenal sejak tahun 1990, ketika pekerjaan pemanfaatan data menjadi sesuatu yang penting dalam berbagai bidang, mulai dari bidang akademik, bisnis, hingga medis (Prasetyo, Eko ; 2014 : 3).

Data Mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. *Data mining* adalah suatu metode pengolahan data untuk menemukan pola yang tersembunyi dari data tersebut. Hasil dari pengolahan data dengan metode *data mining* ini dapat digunakan untuk mengambil keputusan di masa depan. *Data mining* ini juga dikenal dengan istilah *pattern recognition* (Santosa, 2007).

Data mining merupakan metode pengolahan data berskala besar oleh karena itu data mining ini memiliki peranan penting dalam bidang industri, keuangan, cuaca, ilmu dan teknologi. Secara umum kajian data mining membahas metode-metode seperti, *clustering*, *klasifikasi*, *regresi*, *seleksi variable*, dan *market basket analisis* (Santosa, 2007).

Data mining erat kaitanya dengan *machine learning*. *Machine learning* adalah suatu area dalam *artificial intelligence* atau kecerdasan buatan yang berhubungan dengan pengembangan teknik – teknik yang bisa diprogramkan dan belajar dari data masa lalu.

Hubungan yang dicari dalam *data mining* dapat berupa hubungan antara dua atau lebih dalam satu dimensi. Keluaran dari *data mining* bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan berdasarkan informasi yang diperoleh dari data masa lalu (Kusrini, Emha Taufiq Lutfi ; 2009 : 3-5).

Istilah *data mining* memiliki beberapa padanan, seperti *knowledge discovery* atau *pattern recognition*. Kedua istilah tersebut sebenarnya memiliki ketepatannya masing – masing. Istilah *knowledge discovery* atau penemuan pengetahuan tepat digunakan karena tujuan utama dari data mining untuk mendapatkan pengetahuan yang masih tersembunyi didalam bongkahan data. Istilah *pattern recognition* atau pengenalan pola pun tepat untuk digunakan karena pengetahuan yang hendak digali berbentuk pola – pola yang mungkin juga masih perlu digali dari dalam bongkahan data yang tengah dihadapi (Sani Santoso, Dedy Suryadi ; 2010 : 2)

II.1.2. Proses *Knowledge Discovery in Database (KDD)*

Salah satu tahapan dalam keseluruhan proses *knowledge discovery in database (KDD)* adalah data mining. Proses KDD secara garis besar dapat dijelaskan sebagai berikut :

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasioanal perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing/ Cleaning*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang mejadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*).

3. *Transformation*

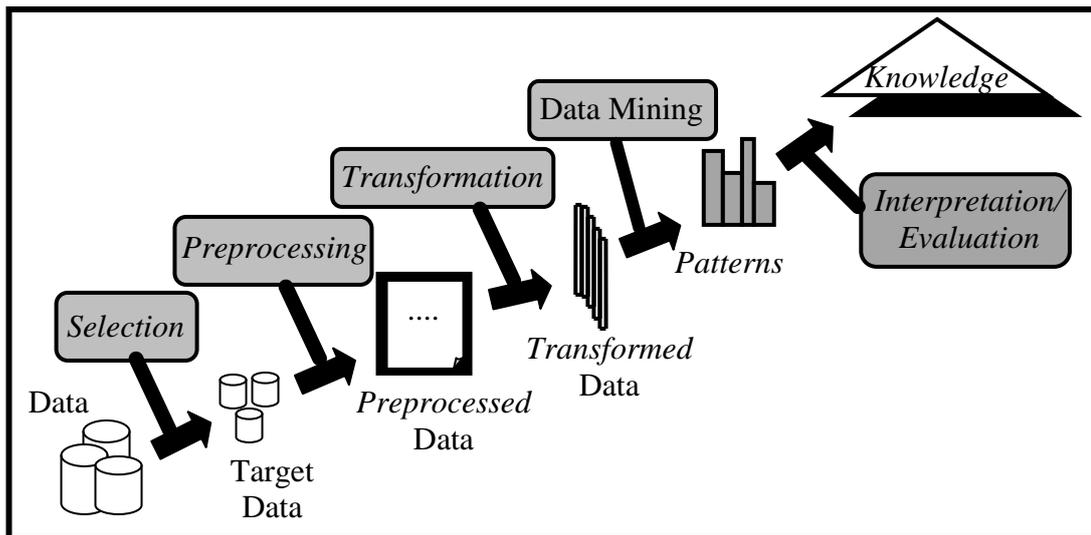
Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data Mining*

Data Mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat tergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation/ Evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya (Kusrini, Emha Taufiq Lutfi ; 2009 : 7).



Gambar II.1. Proses KDD
(Sumber : Hermawati ; 2009 : 6)

II.1.3. Pengelompokan *Data Mining*

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu :

1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. 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 daripada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan untuk prediksi.

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 satu dengan lainnya dan memiliki ketidakmiripan dengan *record – record* dalam kluster lain. Algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok – kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja (Kusrini, Emha Taufiq Lutfi ; 2009 : 10-12).

II.2. Klasterisasi (*Clustering*)

Pada dasarnya *clustering* merupakan suatu metode untuk mencari dan mengelompokkan data yang memiliki kemiripan karakteristik (*similarity*) antara satu data dengan data yang lain. (Santosa, 2007).

Menurut Ediyanto, dkk (2013 : 134), *Cluster Analysis* merupakan salah satu metode objek *mining* yang bersifat tanpa latihan (*unsupervised analysis*).

Analisa kluster atau *clustering* merupakan proses membagi data dalam satu himpunan ke dalam beberapa kelompok yang kesamaan datanya dalam suatu kelompok lebih besar daripada kesamaan data tersebut dengan data dalam kelompok lain (Jang, Sun, dan Mizutani, 2004).

Analisa *cluster* yaitu menemukan kumpulan objek hingga objek-objek dalam satu kelompok sama (atau punya hubungan) dengan yang lain dan berbeda (atau tidak berhubungan) dengan objek-objek dalam kelompok lain. Tujuan dari analisa *cluster* adalah meminimalkan jarak di dalam *cluster* dan memaksimalkan jarak antar cluster. (Astuti Hermawati, Fajar ; 2013 : 123)

Aplikasi dari analisa cluster dibedakan menjadi dua, yaitu :

1. *Understanding*, antara lain kelompok dokumen-dokumen yang saling berhubungan untuk proses browsing, pengelompokan gen dan protein yang mempunyai fungsi sama atau pengelompokan stok dengan harga yang fluktuatif.
2. *Summarization* untuk menurunkan ukuran dari data-set yang besar.

Yang bukan termasuk cluster analisis adalah :

1. *Supervised classification*, yang mempunyai informasi label kelas.

2. *Simple segmentation*, misalkan membagi siswa ke dalam kelompok berdasarkan urutan nama.
3. Hasil dari sebuah *query* yang pengelompokannya merupakan hasil dari spesifikasi eksternal.
4. *Graph partitioning* dimana dalam graph berbeda mempunyai hubungan relevansi dan sinergi tetapi areanya tidak identik. (Astuti Hermawati, Fajar ; 2013 : 124)

Dalam *data mining* ada dua jenis metode *clustering* yang digunakan dalam pengelompokan data, yaitu *hierarchical clustering* dan *non-hierarchical clustering* (Santosa, 2007).

Dalam penulisan skripsi ini penulis menggunakan penerapan metode *non-hierarchical clustering* yaitu yang dimulai dengan menentukan terlebih dahulu jumlah cluster yang diinginkan (dua cluster, tiga cluster, atau lain sebagainya). Setelah jumlah cluster diketahui, baru proses cluster dilakukan tanpa mengikuti proses hierarki. Metode ini biasa disebut dengan *K-Means Clustering* (Santoso, 2010).

II.2.1. Algoritma K- Means Clustering

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

Sedangkan menurut Ediyanto, dkk(2013:134), *K-Means Cluster Analysis* merupakan salah satu atau lebih *cluster* atau kelompok objek berdasarkan karakteristiknya, sehingga objek yang mempunyai karakteristik yang sama dikelompokkan dalam satu *cluster* yang sama dan objek yang mempunyai karakteristik yang sama dikelompokkan dalam suatu *cluster* yang sama dan objek yang mempunyai karakteristik yang berbeda dikelompokkan kedalam *cluster* yang lain. Tujuan pengelompokan adalah untuk meminimalkan *Objective function* yang diset dalam proses *clustering*. Yang pada dasarnya berusaha untuk meminimalkan variasi dalam satu cluster dan memaksimalkan variasi antar *cluster*.

Metode *cluster* ini meliputi *sequential threshold*, *paralel threshold* dan *Optimizing threshold*. *Sequential threshold* melakukan pengelompokan dengan terlebih dahulu memilih satu objek dasar yang akan dijadikan nilai awal *cluster*, kemudian semua *cluster* yang ada dalam jarak terdekat dengan *cluster* ini akan bergabung, lalu dipilih *cluster* kedua dan semua objek yang mempunyai kemiripan dengan *cluster* ini akan digabungkan, demikian seterusnya sehingga terbentuk beberapa *cluster* dengan keseluruhan objek yang terdapat di dalamnya.

Jika diberikan sekumpulan objek $X = (X_1, X_2, \dots, X_n)$ maka algoritma *K-Means Cluster Analysis* akan mempartisi X dalam k buah *Cluster*, setiap *cluster* memiliki *centroid* dari objek-objek dalam *cluster* tersebut. Pada tahap awal algoritma *K-means Cluster Analysis* dipilih secara acak k buah objek sebagai *centroid*, kemudian jarak anatara objek dengan *centroid* dihitung dengan menggunakan jarak *euclidian*, objek ditempatkan dalam *cluster* yang terdekat dihitung dari titik tengah *cluster*. *Centroid* baru ditetapkan jika semua objek sudah

ditempatkan dalam cluster terdekat. Proses penentuan *centroid* dan penempatan objek dalam *cluster* diulangi sampai nilai *centroid konvergen* (centroid dari semua *cluster* tidak berubah lagi).

Adapun, langkah - langkah yang dilakukan dalam melakukan penerapan *k-means clustering* pada data, sesuai dengan ketentuan menurut Santosa(2007) ialah sebagai berikut :

- d. Pilih jumlah *cluster k*.
- e. Inisialisasi *k* pusat *cluster* ini bisa dilakukan dengan berbagai cara. Namun yang paling sering dilakukan adalah dengan cara random. Pusat-pusat *cluster* diberi nilai awal dengan angka-angka random,
- f. Alokasikan semua data/objek ke *cluster* terdekat. Kedekatan dua objek ditentukan berdasarkan jarak kedua objek tersebut. Demikian juga kedekatansuatu data ke *cluster* tertentu ditentukan jarak antara data dengan pusat *cluster*.

Dalam tahap ini perlu dihitung jarak tiap data ke tiap pusat *cluster*. Jarak paling dekat antara satu data dengan satu *cluster* tertentu akan menentukan suatu data masuk dalam *cluster* mana. Untuk menghitung jarak semua data ke setiap titik pusat *cluster* dapat menggunakan teori jarak *Euclidean* yang dirumuskan sebagai berikut:

$$D(i, j) = \sqrt{(x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2 + \dots + (x_{ki} - x_{kj})^2} \dots (1)$$

dimana:

$D(i, j)$ = Jarak data ke *i* ke pusat *cluster j*

x_{ki} = Data ke *i* pada atribut data ke *k*

x_{kj} = Titik pusat ke j pada atribut ke k

- g. Hitung kembali pusat *cluster* dengan keanggotaan *cluster* yang sekarang. Pusat *cluster* adalah rata-rata dari semua data/objek dalam *cluster* tertentu. Jika dikehendaki bisa juga menggunakan median dari *cluster* tersebut. Jadi rata-rata (*mean*) bukan satu-satunya ukuran yang bisa dipakai.
- h. Tugaskan lagi setiap objek memakai pusat *cluster* yang baru. Jika pusat *cluster* tidak berubah lagi maka proses *clustering* selesai. Atau, kembali ke langkah nomor 3 sampai pusat *cluster* tidak berubah lagi.

II.3. PT. Fonterra Brands Indonesia

Susu merupakan bahan makanan yang mempunyai nilai gizi tinggi. Susunan nilai gizi yang sempurna ini merupakan medium yang sangat baik bagi pertumbuhan organisme. (Ace, Supangkat : 2006). Susu merupakan salah satu bahan pangan yang sangat penting bagi pemenuhan kebutuhan gizi masyarakat. Susu berperan sebagai asupan penting untuk kesehatan, kecerdasan, dan pertumbuhan, bagi setiap orang. (Miftah, Heny : 2011)

PT. Fonterra Brands Indonesia (FBI) merupakan perusahaan yang terkenal dengan produk susunya, yang berpusat di New Zealand. Sebagai salah satu yang terdepan di industri susu dunia, Fonterra mengelola sepertiga dari total produksi susu secara global, termasuk di Indonesia. (Marshadita, Feliza, dkk : 2012)

Di New Zealand, Fonterra merupakan salah satu tulang punggung perekonomian negara yang menyerap 97 persen produksi susu di negara tersebut,

sebagian besar diekspor ke 140 pasar di dunia.. Setiap tahun Fonterra memproduksi 2 juta ton susu dan produk turunannya, dimana 80% sudah diproses atau dikemas di berbagai negara. Di wilayah Asia dan Timur Tengah, Indonesia memberikan kontribusi terbesar sekitar 28 persen.

Fonterra merupakan salah satu perusahaan berbasis susu dengan portfolio terlengkap di dunia yang memenuhi kebutuhan susu di setiap tahapan hidup manusia. Di bawah payung Fonterra Brands Nutrition Policy & Guidelines, Fonterra berkomitmen untuk memiliki standar tertinggi dalam berinovasi demi pemenuhan gizi sesuai dengan kebutuhan terkini di masing-masing negara.

Di Indonesia jajaran brands Fonterra adalah Anmum (Anmum Infacare, Anmum Essential, Anmum Materna, Anmum Lacta), Anlene, Boneeto, dan Anchor.

Menurut Direktorat Budidaya Ternak Ruminansia pada tahun 2010 : Permintaan susu tumbuh sangat cepat, dan meningkat 14,01% selama periode antara tahun 2002 dan tahun 2007 (Miftah, Heny : 2011). Peluang pendistribusian susu di Indonesia sangat baik, mengingat Indonesia memiliki jumlah penduduk yang cukup besar. (Lim Sanny : 2010).

II.4. 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.

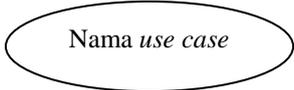
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

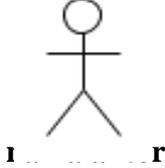
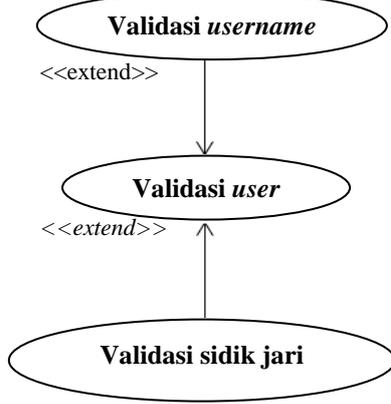
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

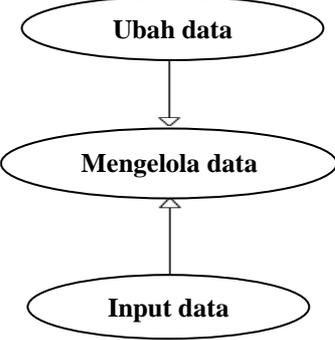
1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*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. Berikut ini adalah simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol – Simbol *Use Case* Diagram

Simbol	Deskripsi
<p data-bbox="331 1518 448 1547"><i>Use case</i></p> 	<p data-bbox="715 1518 1326 1626"><i>Use case</i> merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.</p>

<p>Aktor / <i>actor</i></p> 	<p>Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.</p>
<p>Asosiasi/ <i>association</i></p> 	<p>Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p> <p><<<i>extend</i>>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>

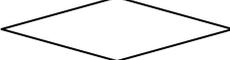
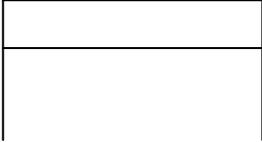
	 <p>Arah panah mengarah pada use case yang menjadi generalisasinya (umum).</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><code><<include>></code> </p> <p><code><<uses>></code> </p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <p><i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</p> <p><i>uses</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.</p>

Sumber : (Rosa A.S, M.Shalahuddin ; 2013 : 155-158).

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Berikut ini adalah simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.2. Simbol - Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane nama swimlane 	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

Sumber : (Rosa A.S, M.Shalahuddin; 2013 : 161-163).

3. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung 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.3. Multiplicity Class Diagram

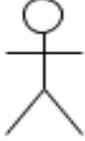
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 9)

4. 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*

Simbol	Deskripsi
<p data-bbox="327 465 406 495">Aktor</p>  <p data-bbox="427 719 587 748">nama aktor</p>	<p data-bbox="715 465 1337 748">orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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 di awal nama <i>actor</i>.</p>
<p data-bbox="327 842 596 871">Garis hidup / <i>lifeline</i></p> 	<p data-bbox="715 842 1182 871">Menyatakan kehidupan suatu objek.</p>
<p data-bbox="327 1149 411 1178">Objek</p> <div data-bbox="327 1178 644 1256" style="border: 1px solid black; padding: 2px;"> <p data-bbox="352 1193 619 1223"><u>nama objek :nama kelas</u></p> </div>	<p data-bbox="715 1149 1275 1178">Menyatakan objek yang berinteraksi pesan.</p>
<p data-bbox="327 1442 485 1471">Waktu aktif</p> 	<p data-bbox="715 1442 1310 1581">Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah tahapan yang dilakukan didalamnya.</p>
<p data-bbox="327 1733 549 1762">Pesan tipe create</p> <p data-bbox="427 1805 592 1834"><<<i>create</i>>></p> 	<p data-bbox="715 1733 1337 1834">Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>

<p>Pesan tipe call</p> <p>1 : nama_metode{ }</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe send</p> <p>1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe return</p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe destroy</p> <p><<destroy>></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>.</p>

Sumber : (Rosa A.S, M.Shalahuddin ; 2013 : 165-167).

II.5. Basis Data

Basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital. Suatu basis data terdiri dari sekumpulan tabel yang saling berelasi ataupun tidak berelasi.

semua tabel tersebut merupakan representasi tempat untuk penyimpanan data yang mendukung fungsi dari basis data tersebut pada suatu sistem (Yudi Priadi ; 2014 : 1-2).

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (redundance) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file./tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

II.5.1. Basis Data Relational

Basis data *relational* diciptakan oleh seorang peneliti dari IBM yang bernama Dr. E. F. Codd, kemudian dikembangkan terus oleh peneliti lainnya hingga saat ini. Prinsipnya, model basis data *relational* digunakan sebagai suatu cara untuk mengelompokkan data dari sebuah kumpulan data yang besar. hal ini dapat dilakukan dengan cara menghapus duplikasi dari suatu data melalui proses yang disebut normalisasi. Proses ini terdiri dari beberapa langkah yang akan menjadi suatu bentuk normal. Hasilnya berupa bahasa umum untuk melakukan akses terhadap suatu basis data, yang disebut dengan *Structured Query*

Language (SQL), sehingga dimungkinkan untuk melakukan *query* terhadap organisasi struktur datanya. SQL sebagai bahasa pada basis data telah menjadi standar untuk semua perusahaan pengembang basis data hingga saat ini (Yudi Priadi ; 2014: 14).

II.6. Kamus Data

II.6.1. Pengertian 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).

Berikut ini adalah contoh kamus data untuk studi kasus Sistem Informasi Manajemen Perpustakaan :

Tabel II.5. Studi Kasus Kamus Data

1	Nama	:	<i>Login</i>
	Digunakan di	:	Proses <i>login</i> (masukan(input))
	Deskripsi	:	<i>Login = id+password</i> <i>Id =*string*</i> <i>Password =*string*</i>
2	Nama	:	Status <i>login</i>
	Digunakan di	:	Proses memeriksa status <i>login</i> (keluaran(<i>output</i>)) Proses mengelola pustaka (masukan(input)) Proses mengelola anggota (masukan(input)) Proses mengelola peminjaman (masukan(input)) Proses mengelola petugas (masukan(input))
	Deskripsi	:	Status <i>login</i> = [ya tidak] Status <i>login</i> = <i>*boolean*</i>
3	Nama	:	Pesan <i>login/logout</i> (pesan untuk ditampilkan ke petugas perpustakaan)
	Digunakan di	:	Proses <i>login/logout</i> (keluaran(<i>output</i>))
	Deskripsi	:	Pesan <i>login/logout</i> = <i>*string*</i>
4	Nama	:	Pencarian pustaka
	Digunakan di	:	Proses mencari pustaka(masukan(input))
	Deskripsi	:	Pencarian=kata kunci +kategori pencarian kata kunci= <i>*string*</i> Kategori = <i>*string*</i>

(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 87)

II.7. Teknik Normalisasi

II.7.1. Pengertian 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 telah ditemukan.

II.7.2. Bentuk-bentuk Normalisasi

1. Bentuk normal tahap pertama (1st 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. Adapun contohnya dapat terlihat pada tabel berikut :

Tabel II.6. Contoh Data Belum Ternormalisasi

Nomor_ Pesanan	Tgl_ Pesanan	Item1	Item2	Item3	Item4	Total (Rupiah)
50001	12/05/1997	P1	P2	P3	P4	45000
50002	12/05/1997	P3	P5	P6		32500
50003	13/05/1997	P1	P2			12000

(Sumber : Abdul Kadir, 2003 : 75)

Tabel II.7. Contoh Normalisasi Bentuk Pertama

Nomor_ Pesanan	Tgl_ Pesanan	Item	Total
50001	12/05/1997	P1	45000
50001	12/05/1997	P2	45000
50001	12/05/1997	P3	45000
50001	12/05/1997	P4	45000
50002	12/05/1997	P3	32500
50002	12/05/1997	P5	32500
50002	12/05/1997	P6	32500
50003	13/05/1997	P1	12000
50003	13/05/1997	P2	12000

(Sumber : Abdul Kadir, 2003 : 75)

2. Bentuk normal tahap kedua (2nd 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.

Adapun contohnya dapat terlihat pada tabel berikut :

Tabel II.8. Contoh Normalisasi Bentuk Kedua

Nomor_Pesanan	Tgl_Pesanan	Total
50001	12/05/1997	45000
50002	12/05/1997	32500
50003	13/05/1997	12000

Nomor_Pesanan	Item
50001	P1
50001	P2
50001	P3
50001	P4
50002	P3
50002	P5
50002	P6
50003	P1
50003	P2

(Sumber : Abdul Kadir, 2003 : 80)

3. Bentuk normal tahap ketiga (3rd 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. Adapun contohnya dapat terlihat pada tabel berikut :

Tabel II.9. Contoh Normalisasi Bentuk Ketiga

No_ Pesanan	Nomor_ Urut	Kode_ Item
50001	0001	P1
50001	0002	P2
50001	0003	P3
50001	0004	P4
50002	0001	P3
50002	0002	P5
50002	0003	P6
50003	0001	P1
50003	0002	P2

Kode_Item	Nama_ Item
P1	Pensil
P2	Buku Tulis
P3	Penggaris
P4	Penghapus
P5	Pulpen
P6	Spidol

(Sumber : Abdul Kadir, 2003 : 83)

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 multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

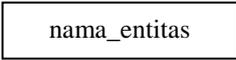
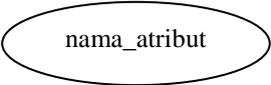
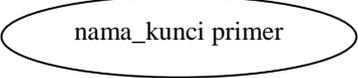
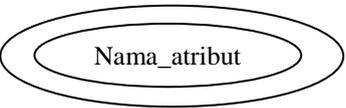
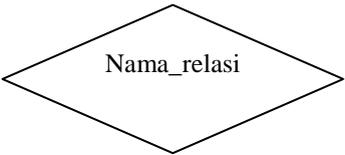
Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi *lossless* menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join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

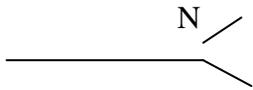
II.8. Entity Relationship Diagram (ERD)

II.8.1. Pengertian Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata ; 2010 : 67).

Tabel II.10. Simbol-Simbol ERD

Simbol	Deskripsi
<p>Entitas / <i>entity</i></p> 	<p>Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.</p>
<p>Atribut</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.</p>
<p>Atribut kunci primer</p> 	<p><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; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).</p>
<p>Atribut multivalai / <i>multivalue</i></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p>
<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p>

<p>Asosiasi / <i>association</i></p> 	<p>Penghubung antara relasi dan entitas dimana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian</p> <p>kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan yang lain disebut kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan B maka.</p>
------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sumber : (Rosa A.S, M.Shalahuddin ; 2013 : 50-51).

II.9. PHP Hypertext Preprocessor (PHP)

Pada tahun 1994 seorang programmer bernama Rasmus Lerdorf awalnya membuat sebuah halaman website pribadi, tujuannya adalah untuk mempertahankan halaman *website* pribadi tersebut sekaligus membangun halaman *web* yang dinamis. PHP pada awalnya diperkenalkan sebagai singkatan dari *Personal Home Page*. PHP pertama ditulis menggunakan bahasa Perl (*Perl Script*), kemudian ditulis ulang menggunakan bahasa pemrograman C CGI-BIN (*Common Gateway Interface-Binary*) yang ditujukan untuk mengembangkan halaman *website* yang mendukung formulir dan penyimpanan data. Pada tahun 1995 *PHP Tool 1.0* dirilis untuk umum, kemudian pengembangannya dilanjutkan oleh Andi Gutmans dan Zeev Suraski. Perusahaan bernama Zend kemudian melanjutkan pengembangan PHP dan merilis PHP versi terakhir pada saat ini. (Sibero, 2013 : 49).

A. Pengertian PHP

Menurut Sibero (2013 : 49), PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. PHP adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhannya.

B. Penulisan Kode PHP

Pemrograman PHP dapat ditulis dalam dua bentuk yaitu penulisan baris kode PHP pada *file* tunggal dan penulisan kode PHP pada halaman html (*embedded*). Kedua cara penulisan tersebut tidak memiliki perbedaan, hanya menjadi kebiasaan gaya penulisan dari programmer. (Sibero, 2013 : 49-50).

Berikut ini adalah contoh penulisan kode program PHP :

- a. Contoh 1 dari penulisan baris kode PHP pada *file* tunggal.

singlefile_php.php

```
<?php
echo "<html>";
echo "<head>";
echo "<title>Contoh PHP</title>";
echo "</head>";
echo "<body>";
echo "<p>Di bawah ini adalah tulisan dari PHP</p>";
echo "Teks dari PHP";
echo "</html>";
?>
```

- b. Contoh 2. Penulisan kode PHP pada halaman html (*embedded*).

Embedded_php.html

```
<html>
  <head>
    <title>Contoh PHP</title>
  </head>
  <body>
    <p>Di bawah ini adalah tulisan dari PHP</p>
    <?php
      echo "Teks dari PHP";
    ?>
  </body>
</html>
```

II.10. *HyperText Markup Language* (HTML)

HyperText Markup Language atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*. Struktur dokumen HTML terdiri dari *tag* pembuka dan *tag* penutup. HTML versi 1.0 dibangun oleh W3C dan terus mengalami perkembangan. Sampai saat ini HTML terakhir adalah versi 5.0. Struktur dokumen HTML sebagai berikut :

```
<html>
  <head>
    <title></title>
  </head>
  <body></body>
</html>
```

Struktur di atas adalah satu kesatuan yang harus ada dalam setiap dokumen HTML. Dokumen HTML terdiri dari komponen yaitu *tag*, elemen dan atribut. (Sibero, 2013 : 19).

II.11. *Cascading Style Sheet (CSS)*

Cascading Style Sheet dikembangkan untuk menata gaya pengaturan halaman *web*. Pada awalnya CSS dikembangkan pada SGML pada tahun 1970 dan terus dikembangkan hingga saat ini CSS telah mendukung banyak bahasa *Markup* seperti : HTML, XHTML, XML, SVG (*Scalable Vector Graphics*) dan Mozilla XUL (*XML User interface Language*). Mengacu dari arti bahasa, *Cascading Style Sheet* memiliki arti Gaya Menata Halaman Bertingkat, yang berarti setiap satu elemen yang telah diformat dan memiliki anak dan telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya. (Sibero, 2013 : 112-113).

Cascading Style Sheet terdiri dari *Selector*, Properti dan Nilai. Seperti halnya HTML, PHP dan bahasa pemrograman lainnya, CSS juga memiliki aturan penulisan. Berikut di bawah ini aturan penulisan CSS :

```
span // span => Selector
{
    font-weight: bold; // font-weight => properti, bold => nilai
    color: red; // color => properti, red => nilai
}
```

II.12. MySQL

MySQL atau dibaca “*My Sekuel*” dengan adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. MySQL pertama kali dikembangkan oleh MySQL AB yang kemudian diakuisisi Sun Microsystem dan terakhir dikelola oleh Oracle Corporation. (Sibero, 2013 : 97).

Menurut Nugroho, MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar *SQL (Structured Query Language)*. MySQL memiliki dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. MySQL yang biasa kita gunakan adalah *MySQL FreeSoftware* yang berada di bawah Lisensi *GNU/GPL (General Public Licence)*.

II.13. Adobe Dreamweaver

Adobe Dreamweaver adalah suatu produk *Web Developer* yang dikembangkan oleh *Adobe Systems Inc.*, sebelumnya produk *Dreamweaver* dikembangkan oleh *Macromedia Inc.* yang kemudian sampai dengan saat ini pengembangannya diteruskan oleh *Adobe Systems Inc.* Setelah diambil alih oleh *Adobe Systems Inc.*, *Dreamweaver* dikembangkan dan dirilis dengan kode nama *Creative Suite (CS)*.

Ruang Kerja atau *Workspace* adalah bagian keseluruhan tampilan dari *Adobe Dreamweaver*. Ruang kerja *Dreamweaver* terdiri dari *Welcome Screen*, *Menu*, *Insert bar*, *Document Window*, *CSS Panel*, *Application Panel*, *Tag Inspector*, *Property Inspector*, *Result Panel* dan *Files Panel*. Masing-masing dari komponen tersebut memiliki fungsi dan aturan. (Sibero, 2013 : 384).