

BAB II

LANDASAN TEORI

2.1 Data Mining

Data Mining (DM) adalah salah satu bidang yang berkembang pesat karena besarnya kebutuhan akan nilai tambah dari database skala besar yang makin banyak terakumulasi sejalan dengan pertumbuhan teknologi informasi. Definisi umum dari DM itu sendiri adalah serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data.

Menurut Abdul Kadir (2013:30), Data mining adalah kombinasi secara logis antara pengetahuan data, dan analisa statistik yang dikembangkan dalam pengetahuan bisnis atau suatu proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, tiruan dan *machine-learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat bagi pengetahuan yang terkait dari berbagai database besar.

Menurut Rama Novta Miraldi, dkk dalam penelitiannya *Implementasi Algoritma FP-Growth untuk sistem Rekomendasi Buku di perpustakaan UKDW*. Dengan memberikan feature perekomendasi relasi buku yang sedang dicari. Feature ini membantu pengguna untuk menemukan literatur atau topik yang berhubungan. Algoritma FP-Growth dapat digunakan dalam kasus perekomendasi buku dan mendapatkan hasil keluaran berupa rekomendasi buku yang terkait, selain itu tingkat keakuratan FP-Growth dalam memberikan rekomendasi buku adalah sebesar 60,78% (Rama Novta Miraldi, 2014).

Penelitian yang dilakukan Erwin, melakukan analisis market basket dengan algoritma Apriori dan FP-Growth menyimpulkan bahwa algoritma Apriori membutuhkan waktu komputasi yang lama untuk mendapatkan frequent itemsets, karena berulang kali melakukan pemindahan data dan membutuhkan memori yang besar sedangkan FP-Growth hanya membutuhkan dua kali scanning database dalam mencari frequent itemsets sehingga waktu yang dibutuhkan menjadi relatif singkat dan efisien (Erwin, 2009).

Penelitian yang lain dengan judul *Penerapan Data Mining Untuk Analisis Pola Pembelian Konsumen dengan Algoritma FP-Growth pada Data Transaksi Penjualan Spare Part Motor*. Dengan dapat mengetahui secara cepat hasil penjualan dari spare part motor sport yang paling banyak terjual pada perusahaan yang mempunyai banyak cabang atau dealer. Akan memberikan keuntungan kepada pemilik usaha serta dengan cepat menentukan strategi penjualan (Alfannisa Annurullah Fajrin, 2018).

Penelitian terkait algoritma FP-Growth yang lain yaitu mengenai *Penentuan Pola Yang Sering Muncul Untuk Penerima Kartu Jaminan Kesehatan Masyarakat (Jamkesmas) Menggunakan Metode Fp-Growth*. Ketidaktepatan penilaian tim dalam memberikan penilaian terhadap pasien JAMKESMAS merupakan sumber masalah utama. Dengan menggunakan Algoritma Fp-Growth analisa data yang dihasilkan memperoleh pengetahuan pola yang sering muncul pada penerima JAMKESMAS di kelurahan Bambe kecamatan Driyorejoyang tepat sasaran masih 60% dikarenakan masih banyak penduduk yang kurang memenuhi kriteria (Meilani, 2015).

Menurut Arincy (2014) dalam penelitian berjudul “*Multidimensional Association Rules Mining* untuk Data Kebakaran Hutan Menggunakan Algoritme FP-Growth dan *ECLAT*”, dinyatakan bahwa dari hasil pengujian algoritma Apriori, FP-Growth, dan *ECLAT* pada penelitian ini dihasilkan aturan asosiasi yang sama, baik jumlah *frequent itemset*-nya maupun jumlah aturan asosiasi yang dihasilkan. Walaupun proses membangkitkan *frequent itemset* ketiga algoritma ini berbeda, jumlah *frequent itemset* yang dihasilkan tetap sama. Hal tersebut disebabkan karena jumlah transaksi yang diproses belum terlalu banyak.

Dan Toomey (dalam Arinda, Dwi Syafina, Sulastri. 2017:3) mengatakan bahwa, Algoritma *ECLAT* digunakan untuk *frequent itemset mining*. Algoritma ini menggunakan persimpangan dalam data untuk menghitung dukungan kandidat.

Menurut Samodra (2015) dalam penelitian berjudul “Implementasi Algoritma *ECLAT* untuk *Frequent Pattern Mining* pada Penjualan Barang”, disimpulkan bahwa hasil perbandingan pola penjualan pada tahun 2011 dan 2012 memiliki nilai yang relatif sama pada nilai *support* dan nilai *lift*-nya. Algoritma *ECLAT* ini mampu melakukan proses analisa data dengan waktu relatif singkat (2304 transaksi dalam waktu 61ms dan 1984 transaksi dalam waktu 19ms pada *minimum support* 10%).

Data mining meliputi tugas-tugas yang dikenal sebagai ekstraksi pengetahuan, arkeologi data, eksplorasi dalam pemrosesan pola data dan memanen informasi. Semua aktifitas ini dilakukan secara otomatis dan mengizinkan adanya penemuan cepat pengetahuan baru bahkan oleh non programmer. *Data mining* cerdas menemukan informasi di dalam data warehouse

dimana laporan dan *query* tidak bisa diungkapkan secara efektif. Piranti data mining menemukan pola-pola di dalam data dan bahkan menyimpulkan aturan dari data tersebut Definisi umum dari Data Mining adalah proses pencarian pola-pola yang menarik (hidden pattern) berupa pengetahuan.

Knowledge yang tidak diketahui sebelumnya dari suatu kumpulan data dimana data tersebut dapat berada dalam database, *data warehouse*, atau media penyimpanan informasi yang lain. Data Mining merupakan proses analisis terhadap data dengan penekanan menemukan informasi yang tersembunyi pada sejumlah besar data yang disimpan ketika menjalankan bisnis perusahaan. Sumber : (Data Mining dan Web Mining) (andyku).

Pada dasarnya data mining berhubungan erat dengan analisis data dan penggunaan perangkat lunak untuk mencari pola dan kesamaan dalam sekumpulan data. Ide dasarnya adalah menggali sumber yang berharga dari suatu tempat yang sama sekali tidak diduga, seperti perangkat lunak data mining mengekstraksi pola yang sebelumnya tidak terlihat atau tidak begitu jelas sehingga tidak seorang pun yang memperhatikan sebelumnya. Analisa data mining berjalan pada data yang cenderung terus membesar dan teknik terbaik yang digunakan kemudian berorientasi kepada data berukuran sangat besar untuk mendapatkan kesimpulan dan keputusan paling layak. Data mining memiliki beberapa sebutan atau nama lain yaitu : *knowledge discovery in database* (KDD), ekstraksi pengetahuan (knowledge extraction), analisa data / pola (data / pattern analysis), kecerdasan bisnis (business intelligence), data *archaeology* dan data *dredging* (Larose, 2005).

2.1.1 Metode Data Mining

Secara garis besar, Han dalam bukunya menjelaskan bahwa metode data mining dapat dilihat dari dua sudut pandang pendekatan yang berbeda, yaitu pendekatan deskriptif dan pendekatan prediktif [8]. Pendekatan deskriptif adalah pendekatan dengan cara mendeskripsikan data inputan. Metode yang termasuk ke dalam pendekatan ini adalah:

1. Metode deskripsi konsep/kelas, yaitu data dapat diasosiasikan dengan kelas atau konsep. Ada tiga macam pendeskripsian yaitu (1) karakteristik data, dengan membuat summary karakter umum atau fitur data suatu kelas target, (2) diskriminasi data, dengan membandingkan class target dengan satu atau sekelompok kelas pembanding, (3) gabungkan antara karakterisasi dan diskriminasi.
2. Metode association rule, yaitu menemukan aturan asosiatif atau pola kombinasi dari suatu item yang sering terjadi dalam sebuah data. Pendekatan kedua adalah pendekatan prediktif, yaitu pendekatan yang dapat digunakan untuk memprediksi, dengan hasil berupa kelas atau cluster.
3. Metode klasifikasi dan prediksi, yaitu metode analisis data yang digunakan untuk membentuk model yang mendeskripsikan kelas data yang penting, atau model yang memprediksikan trend data. Klasifikasi digunakan untuk memprediksi kelas data yang bersifat kategorial, sedangkan prediksi untuk memodelkan fungsi yang mempunyai nilai kontinu.
4. Metode clustering, mengelompokkan data untuk membentuk kelas-kelas baru atau sering disebut cluster. Metode clustering bertujuan untuk

memaksimalkan persamaan dalam satu cluster dan meminimalkan perbedaan antar cluster.

2.1.2 Tahapan Data Mining

Menurut Syaifullah (2010:15), Dalam aplikasinya, data mining sebenarnya merupakan bagian dari proses *Knowledge Discovery in Database* (KDD), bukan sebagai teknologi yang utuh dan berdiri sendiri. Data mining merupakan suatu bagian langkah yang penting dalam proses KDD terutama berkaitan dengan ekstraksi dan dari data yang ditelaah.

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional 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 menjadi fokus KDD. Proses cleaning mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Juga dilakukan proses enrichment, yaitu “memperkaya” data yang sudah ada dengan data informasi atau informasi lain yang relevan.

3. 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 bergantung pada tujuan dan proses KKD secara keseluruhan.

4. 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 KKD yang disebut dengan interpretation. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

2.1.3 Metode Asosiasi

Salah satu pengaplikasian dari *Association Rule* adalah *Market Basket Analysis*, dimana bertujuan untuk menemukan bagaimana item yang dibeli oleh pelanggan dalam supermarket atau toko saling berhubungan. Pencarian *Association Rules* dilakukan melalui dua tahap yaitu pencarian *frequent itemset* dan penyusunan rules. Penting tidaknya suatu *Association Rules* dapat diketahui dengan dua parameter, yaitu *support* (nilai penunjang) dan *confidence* (nilai kepastian). Support adalah ukuran yang menunjukkan tingkat dominasi itemset dari keseluruhan transaksi. Persamaan I untuk menentukan nilai suatu support.

Ada 2 (dua) proses utama dalam penggalian aturan asosiasi, yaitu :

pencarian pola (*frequent pattern*) dari sejumlah transaksi penentuan kuatnya rule (aturan) dari pola yang dihasilkan. Dalam *association rule mining* ada 2 (dua) hal yang mempengaruhi:

1. Support

Support adalah proporsi suatu item dalam semua transaksi. Support dirumuskan sebagai berikut:

$$\text{Support}(A) = \frac{\sum \text{Jumlah Transaksi Mengandung } A}{\sum \text{Jumlah Transaksi}} \times 100\%$$

$$\text{Support}(A, B) = \frac{\sum \text{Jumlah Transaksi Mengandung } A \text{ \& B}}{\sum \text{Jumlah Transaksi}} \times 100\%$$

2. Confidence

Confidence adalah hal yang mendasari aturan asosiasi, dengan konsep implikasi ($x \Rightarrow y$), atau "if ... then ...". Besarnya nilai confidence suatu aturan (rule) dirumuskan sebagai berikut:

$$\text{Conf}(A \Rightarrow B) = \frac{\sum \text{Jumlah Transaksi Mengandung } A \text{ \& B}}{\sum \text{Jumlah Transaksi}} \times 100\%$$

2.1.4 FP-Tree

Frequent Pattern Tree (FP-Tree) adalah representasi pemasukan data yang dipadatkan (Kumar, 2004). FP-Tree dibentuk dengan membaca kumpulan data pada suatu transaksi dalam suatu waktu dan memetakan transaksi tersebut ke dalam lintasan FP-Tree. Transaksi yang berbeda dapat memiliki items yang sama, sehingga memungkinkan lintasannya saling menimpa. Semakin banyak lintasan yang menimpa satu sama lain, dengan menggunakan struktur FP-Tree, maka proses pemadatan akan semakin baik. Dalam, "Mining Frequent Patterns without Candidate Generation: A Frequent Pattern Tree Approach" (Han, Pei, Yin, & Mao, 2004) menjelaskan misal: $\{dt, az, \dots, a\}$ adalah kumpulan

dari item, dan database transaksi DB : $\{T_1, T_2, \dots, T_n\}$, dimana T_i ($i \in [1..n]$) adalah sekumpulan transaksi yang mengandung item di I. Sedangkan support adalah penghitung (counter) frekuensi kemunculan transaksi yang mengandung suatu pola. Suatu pola dikatakan sering muncul (frequent pattern) apabila support dari pola tersebut tidak kurang dari suatu konstanta minimum support (batas ambang minimum support) yang telah di definisikan sebelumnya. Permasalahan mencari pola frequent dengan batas ambang minimum support count, inilah yang dicoba untuk dipecahkan oleh FP-Growth dengan bantuan struktur FP-tree.

2.2 Algoritma FP-Growth

Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algoritma Apriori diperbaiki oleh algoritma FP-Growth. Frequent Pattern Growth (FP-Growth) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (frequent itemset) dalam sebuah kumpulan data. Pada algoritma Apriori diperlukan generate candidate untuk mendapatkan frequent itemsets. Akan tetapi, di algoritma FP-Growth generate candidate tidak dilakukan karena FP-Growth menggunakan konsep pembangunan tree dalam pencarian frequent itemsets. Hal tersebutlah yang menyebabkan algoritma FP-Growth lebih cepat dari algoritma Apriori. Karakteristik algoritma FP-Growth adalah struktur data yang digunakan adalah tree yang disebut dengan FP-Tree. Dengan menggunakan FP-Tree, algoritma FP-growth dapat langsung mengekstrak frequent Itemset dari FP- Tree.

Metode FP-Growth dapat dibagi menjadi 3 tahapan utama yaitu sebagai (Erwin, 2009):

1. Tahap pembangkitan *conditional pattern base*

Conditional Pattern Base merupakan subdatabase yang berisi prefix path (lintasan prefix) dan suffix pattern (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui FP-Tree yang telah dibangun sebelumnya.

2. Tahap pembangkitan conditional FP-Tree

Pada tahap ini, *support count* dari setiap item pada setiap conditional pattern base dijumlahkan, lalu setiap item yang memiliki jumlah support count lebih besar sama dengan minimum *support count* akan dibangkitkan dengan Conditional FP-Tree.

3. Tahap pencarian frequent itemset

Apabila Conditional FP-Tree merupakan lintasan tunggal (single path), maka didapatkan frequent itemset dengan melakukan kombinasi item untuk setiap Conditional FP-Tree.

2.3 Algoritma *ECLAT*

Algoritma *ECLAT* (*Equivalence Class Transformation*) digunakan untuk melakukan *itemset mining*. Penambangan *itemset* untuk menemukan pola yang sering terjadi pada data seperti jika konsumen membeli roti, ia juga membeli mentega. Jenis pola ini disebut aturan asosiasi dan digunakan di banyak domain aplikasi.

Dan Toomey (dalam Arinda, Dwi Syafina, Sulastri. 2017:3) mengatakan bahwa, Algoritma *ECLAT* digunakan untuk *frequent itemset mining*. Algoritma ini menggunakan persimpangan dalam data untuk menghitung dukungan kandidat.

Menurut Samodra et al. (2015) dalam Jurnal Teknologi yang berjudul “Implementasi Algoritma *ECLAT* untuk *Frequent Pattern Mining* pada Penjualan Barang”, dijelaskan bahwa algoritma *ECLAT* adalah sebuah program yang digunakan untuk menemukan *itemset* yang paling sering keluar. Algoritma ini melakukan pencarian *depth first* pada kisi bagian dan menentukan *support* dari *itemset* dengan cara memotong daftar transaksi.

Tabel 2.1. *Vertical Data Format of The Transaction Data* (Han, 2012)

<i>Itemset</i>	TID_set
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

Tabel 2.1. di atas menampilkan *dataset* dalam bentuk *vertical data format* untuk setiap 1 *itemset* yang berarti, tabel ini menampilkan transaksi pada setiap *item*.

Tabel 2.2. *2-Itemsets in Vertical Data Format of The Transaction Data* (Han, 2012)

<i>Itemset</i>	TID_set
I1, I2	{T100, T400, T800, T900}
I1, I3	{T500, T700, T800, T900}
I1, I4	{T400}
I1, I5	{T100, T800}

I2, I3	{T300, T600, T800, T900}
I2, I4	{T200, T400}
I2, I5	{T100, T800}
I3, I5	{T800}

Tabel 2.2. di atas menampilkan *dataset* dalam bentuk *vertical data format* untuk setiap 2 *itemset* yang berarti, tabel ini menampilkan transaksi pada setiap 2 kombinasi *item*.

Tabel 2.3. 3-Itemsets in Vertical Data Format of The Transaction Data (Han, 2012)

<i>Itemset</i>	TID_set
I1, I2, I3	{T800, T900}
I1, I2, I5	{T100, T800}

Tabel 2.3. di atas menampilkan *dataset* dalam bentuk *vertical data format* untuk setiap 3 *itemset* yang berarti, tabel ini menampilkan transaksi pada 3 setiap kombinasi *item*.

Terdapat sebuah cara mudah untuk merepresentasikan transaksi untuk Algoritma *ECLAT* yaitu dengan menggunakan matriks bit, di mana setiap baris sesuai dengan *item* masing-masing kolom untuk transaksinya (ataupun sebaliknya). Bit adalah set di dalam matriks ini jika *item* sesuai dengan baris yang terkandung di dalam transaksi yang sesuai dengan kolom, selain itu akan dihapus.

2.4 Software Untuk Membangun Aplikasi Web

Untuk membangun aplikasi web pada perancangan Sistem Rekomendasi Peminjaman Buku Pada Perpustakaan SMK Ki Hajar Dewantara Menggunakan Metode *Association Rule* Dengan Algoritma *FP-Growth* Dan Algoritma *ECLAT* penulis menggunakan software sebagai berikut:

1. Xampp
2. Text Editor
3. Mozila Firefox

2.4.1 Aplikasi Website Dan Kelebihannya

Aplikasi web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti internet atau intranet. Ia juga merupakan suatu aplikasi perangkat lunak computer yang dikodekan dalam Bahasa yang didukung penjelajah web (seperti ASP, Perl, Java, Java Script, PHP, Python, Ruby, dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Kelebihan aplikasi web adalah akses informasi mudah, setup server lebih mudah, informasi mudah didistribusikan, bebas platform.

2.4.2 Script Pada Aplikasi Web

2.4.2.1 HTML

Menurut Didik Setiawan (2017:16), “HTML (*Hypertext Markup Language*) adalah sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman website yang dapat diakses atau ditampilkan menggunakan *Web Browser*.”

Menurut Much Aziz Muslim dan Atikah Ari Pramesti dalam *Scientific Journal of Informatics* (2014), “HTML adalah sebuah bahasa untuk menampilkan halaman sebuah website. Html merupakan singkatan dari *Hypertext Markup Language*.

Menurut R.H. Sianipar (2015), “HTML (*Hypertext Mark up Language*) adalah sebuah mark up untuk menstrukturkan dan menampilkan isi dari halaman web.

Berdasarkan beberapa definisi maka bisa disimpulkan bahwa, HTML (*Hypertext Mark up Language*) merupakan script pemrograman standar yang kegunaannya untuk membuat sebuah website agar dapat dijelajahi dengan leluasa didunia maya.

2.4.2.2 CSS

Menurut Johni S Pasaribu dalam *Jurnal Ilmiah Teknologi Informasi Terapan* (2017:158), “CSS dalah singkatan dari *Cascading Style Sheet* yang merupakan kumpulan perintah yang dibentuk dari berbagai sumber yang disusun menurut urutan tertentu sehingga mampu mengatasi konflik style. CSS atau yang disebut *cascading style sheet* yaitu salah satu bahasa pemrograman web yang mengatur komponen dalam suatu web supaya lebih terstruktur dan lebih seragam.

Menurut Didik Setiawan (2017:166), “CSS adalah kependekan dari *Cascading Style Sheet*. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan atau layout halaman web supaya lebih elegan dan menarik.

Berdasarkan pendapat diatas dapat disimpulkan bahwa CSS (*Cascading Style Sheet*) merupakan script. Program yang kegunaanya untuk memberikan gaya tampilan yang menarik dihalaman HTML (*Hypertext Mark up Language*).

2.4.2.3 Java Script

Java Script adalah kode untuk menyusun halaman web yang memungkinkan dijalankan di sisi klien (pada browser yang digunakan pemakai). Karena dijalankan di sisi klien, maka JavaScript digunakan untuk membuat tampilan lebih dinamis. (Abdul Kadir, 2013).

2.4.2.4 PHP

Menurut Budi Raharjo (2016:38), “PHP adalah salah satu bahasa pemrograman script yang dirancang untuk membangun aplikasi web”.

Menurut Jubilee *Enterprise* (2015:1), “PHP singkatan dari *hypertext preprocessor* adalah bahasa komputer atau bahasa pemrograman atau koding atau script yang digunakan untuk mengolah data dari server untuk ditampilkkan di website”.

Menurut Priyo Sutopo dkk dalam Jurnal Infomatika Mulawarman (2016:25), “PHP adalah salah satu server side yang dirancang khusus untuk aplikasi web. PHP disisipkan diantara bahasa HTML dan karena bahasa *server side*, maka bahasa PHP akan dieksekusi di server, sehingga yang dikirimkan ke browser adalah hasil jadi dalam bentuk html, dan kode PHP tidak akan terlihat. PHP termasuk *open source product*. Jadi, dapat diubah source code dan mendistribusikannya secara bebas.

Berdasarkan beberapa definisi diatas dapat disimpulkan bahwa, PHP adalah bahasa pemrograman yang berbentuk script HTML yang prosesnya terjadi pada client-server dan bersifat open source.

2.4.3 Database

Basis data merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi (Indrajani,2014:2). Basis data digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer. Untuk mengelolah database diperlukan suatu perangkat lunak yang disebut DBMS (*Database Management System*).

2.4.3.1 Konsep DBMS (*Database Management System*).

Database Management System (DBMS) merupakan suatu perangkat lunak yang ditujukan untuk menangani penciptaan, pemeliharaan, dan pengendalian akses data (Idam, 2014: 37) . DBMS terdiri dari Database dan set program pengolah untuk menambah data, mengolah data, mengambil dan membaca data. Database sendiri merupakan kumpulan data yang pada umumnya menggambarkan aktifitas dari pelakunnya (*user*). Dalam penelitian ini, penulis menggunakan DBMS berupa MySQL sebagai implementasi perancangan database.

2.4.3.2 MySQL

Menurut Priyo Sutopo dkk dalam Jurnal Informatika Mulawarman (2016:25), "MySQL adalah salah satu jenis *database* server yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

Menurut Mukhamad Masrur (2016:124), mengatakan bahwa "MySQL adalah salah satu *relation database management system* yang bersifat *open source*".

Menurut Budi Raharjo (2015:16), "MySQL adalah software RDBMS (server database) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*) dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Berdasarkan pendapat yang dikemukakan diatas, maka dapat disimpulkan bahwa "MySQL adalah *software* yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar yang diletakan pada beberapa *table* yang terpisah".

Beberapa alasan menggunakan MySQL sebagai *server database* dalam pengembangan sistem adalah:

1. Fleksibel, MySQL memiliki fleksibilitasterhadap terknologi yang akan digunakan sebagai pengembangan aplikasi.

2. Performa tinggi, sebagai database dari beberapa aplikasi web yang memiliki *traffic* yang tinggi.
3. Lintas *Platform*, MySQL dapat digunakan pada sistem operasi yang beragam.
4. Gratis, MySQL dapat digunakan secara gratis.

Proteksi data yang handal, MySQL menyediakan mekanisme yang *powerfull* untuk menangani hal ini, yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data dan lain sebagainya.

2.5 Pemodelan Sistem dengan UML (Unified Modeling Language)

Menurut Onu, Fergus U. dan Umeakuka, Chinelo V. dalam *International Journal of Computer Applications Technology and Research* (2016:506), “A UML is a standard modeling language to model the real world in the field of software engineering. A UML diagram is a partial graphical view of model of a system under design, implementation, or already in existence.

Menurut Ary Budi Warsito dkk dalam Jurnal CCIT (2015:29), “UML (Unified Modelling Language) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut.

Berdasarkan pendapat yang telah dikemukakan oleh para ahli, penulis dapat menarik kesimpulan bahwa UML (Unified Modelling Language) adalah suatu kumpulan pemodelan yang digunakan untuk menentukan atau menggambarkan atau mendokumentasikan sebuah sistem software yang terkait dengan objek.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.

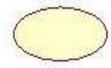
2. Sarana komunikasi antara peangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan system yang ada, proses-proses dan organisasinya.

2.5.1 Usecase Diagram

1. Tabel Komponen *Use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan – persyaratan yang harus dipenuhi sistem dari pandangan pemakai. Beberapa komponen dalam *use case* antara lain:

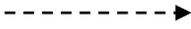
Tabel 2.4 Komponen *Use Case Diagram* (Rosa dan Shalahuddin, 2014:156)

Komponen Use case	Penjelasan
 Aktor	Merupakan sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem.
 <i>Use case</i>	Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Ada beberapa relasi yang terdapat dalam use case, antaran lain :

Tabel 2.5 Relasi Tabel *Use Case* (Rosa dan Shalahuddin, 2014:156)

Relasi <i>Use case</i>	Penjelasan
Relasi <i>Association</i> 	<i>Association</i> , menghubungkan link antar element

Relasi <i>Generalization</i> 	<i>Generalization</i> disebut juga <i>inheritance</i> (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
Relasi <i>Dependency</i> 	<i>Dependency</i> , sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.

Tipe relasi / *stereotype* yang mungkin terjadi pada *use case diagram*:

Tabel 2.6 Tabel *Stereotype* yang mungkin terjadi (Rosa dan Shalahuddin, 2014:156)

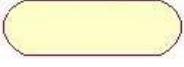
Relasi/ <i>Stereotype</i>	Penjelasan
<< <i>include</i> >>	Kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.
<< <i>extends</i> >>	Kelakuan yang hanya berjalan dibawah kondisi tertentu seperti menggerakkan alarm.
<< <i>Communicates</i> >>	Ditambahkan untuk asosiasi yang mungkin menunjukkan asosiasinya adalah <i>communicates association</i> .

2.5.2 Activity Diagram

1. Tabel Activity Diagram

Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Dapat juga digunakan untuk menggambarkan aliran kejadian dalam *use case*. Berikut beberapa komponen yang terdapat dalam *Activity Diagram*:

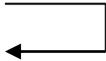
Tabel 2.7 *Activity Diagram* (Rosa dan Shalahuddin, 2014:156)

<i>Activity Diagram</i>	Penjelasan
 <i>Start State</i>	<i>Start State</i> , Sebagai tanda awal proses dari <i>activity diagram</i> .
 <i>State</i>	<i>State</i> , Berfungsi menampung event dalam <i>activity diagram</i> .
 <i>Activity</i>	<i>Activity</i> , Memiliki fungsi yang sama dengan <i>state</i> . Menampung event atau aktifitas pada proses sistem.
 <i>State Transition</i>	<i>State Transition</i> , Berfungsi untuk menunjukkan aliran atau urutan dari event atau aktifitas pada diagram.
 <i>Transition to self</i>	<i>Transition to Self</i> , Berfungsi untuk menunjukkan transisi sebuah event yang mengarah ke event itu sendiri.
 <i>Horizontal Synchronization</i>	<i>Horizontal Synchronization</i> , Berfungsi untuk mengsinkronisasikan 2 cabang event yang posisinya horizontal.
 <i>Decision</i>	<i>Decision</i> , Digunakan ketika terjadi pemilihan 2 kondisi event pada diagram.
 <i>End State</i>	<i>End State</i> , Sebagai tanda akhir dari <i>activity diagram</i> .

2.5.3 Sequence Diagram

Diagram sekuensial umumnya digunakan untuk menggambarkan suatu skenario atau urutan langkah – langkah yang dilakukan baik oleh *actor* maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil atau output. Berikut adalah beberapa komponen yang terdapat dalam *sequence diagram*:

Tabel 2.8 *Sequence Diagram* (Rosa dan Shalahuddin, 2014:156)

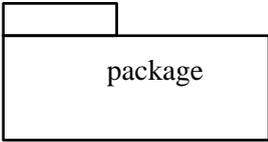
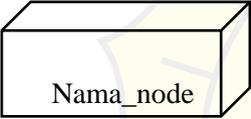
<i>Sequence Diagram</i>	Penjelasan
 <i>Actor</i>	<i>Actor</i> , Menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem.
 <i>Return Message</i>	<i>Return Message</i> , menggambarkan pesan atau hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.
 <i>Lifeline</i>	<i>Lifeline</i> , Eksekusi obyek selama <i>sequence</i> (<i>message</i> dikirim atau diterima dan aktifasinya)
 <i>Message to Self</i>	<i>Message to Self</i> , Menggambarkan pesan atau hubungan obyek itu sendiri yang menunjukkan urutan kejadian yang terjadi.
 <i>Object Message</i>	<i>Object Message</i> , Menggambarkan pesan atau hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.

2.5.4 Deployment Diagram

Deployment diagram adalah diagram yang digunakan memetakan software ke processing node. Menunjukkan konfigurasi elemen pemroses pada saat run

time dan software yang ada di dalamnya. Diagram ini adalah salah satu diagram paling penting dalam tingkat implementasi perangkat lunak. Berikut adalah beberapa komponen yang terdapat dalam *deployment diagram*:

Tabel 2.9 *Deployment Diagram*

<i>Deployment Diagram</i>	Penjelasan
<p data-bbox="300 667 408 701"><i>Package</i></p> 	<p data-bbox="670 667 1366 763"><i>Package</i>, merupakan sebuah bungkus dari satu atau lebih node.</p>
<p data-bbox="300 936 376 969"><i>Node</i></p> 	<p data-bbox="670 936 1366 1305">Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistensikan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.</p>
<p data-bbox="300 1344 647 1377">Kebergantungan/<i>dependency</i></p> 	<p data-bbox="670 1344 1366 1440">Kebergantungan antar node, arah panah mengarah pada node yang dipakai.</p>
<p data-bbox="300 1545 360 1579"><i>Link</i></p> 	<p data-bbox="670 1545 879 1579">Relasi antar node</p>