

BAB II

LANDASAN TEORI

II.1 Penelitian Terkait

Penelitian terkait bertujuan untuk mengambil beberapa referensi jurnal terkait yang digunakan dalam mendukung penelitian publikasi ilmiah dalam jurnal lokal.

Penelitian yang dilakukan oleh Agustin Nina (2018) dengan judul Penerapan Data Mining Metode Apriori Dalam Implementasi Penjualan Di *Alfamart*. Berdasarkan hasil perhitungan dan analisis penelitian konsumen menggunakan Data Mining dengan metode *Algoritma Apriori*, secara keseluruhan dari data sampel penjualan diperoleh 142 *rule* asosiasi yang memenuhi *support* diatas 10% dan *confidence* 50%. Berdasarkan hasil pengujian dengan menggunakan *Rapidminer* dapat disimpulkan bahwa produk yang lebih banyak diminati oleh konsumen adalah minimum *confidence* diatas 50%.

Penelitian yang dilakukan oleh Erfian Junianto (2020) dengan judul Penerapan Data Mining Metode Apriori Dan *Fp-Tree* Pada Penjualan Media Edukasi (Studi Kasus : *Oisha Smartkids*). Berdasarkan hasil analisa dan pengujian pada transaksi penjualan media edukasi menggunakan data mining dengan *algoritma apriori* dari 30 data produk, 12 transaksi setiap bulannya selama tahun 2019 menghasilkan nilai minimum support = 25%, nilai minimum *confidence* 90% dan pola kombinasi produk dan rules sebesar 100%. Selanjutnya dilengkapi dengan

algoritma *FP-tree* menghasilkan 10 produk *best seller* melalui tahap filterisasi dan menemukan pola kombinasi produk.

Penelitian yang dilakukan oleh Rahmad Alfian Syahputra (2020) dengan judul Sistem Informasi Penjualan Pupuk Menggunakan Metode Apriori (Studi Kasus PT. Pupuk Iskandar Muda). Dalam penelitian ini dijelaskan bahwa Aplikasi penjualan pupuk pada PT. Pupuk Iskandar Muda ini membuat proses pemesanan produk dari distributor dapat diakses dimanapun dan kapanpun dikarenakan sistem ini sudah berjalan secara *online*. Sistem informasi penjualan pupuk PT. Pupuk Iskandar Muda ini menjadikan penjualan terkomputerisasi baik dalam proses transaksi, permintaan produk, persetujuan, dan lainnya.

Penelitian yang dilakukan oleh Natalia Silalahi (2020) dengan judul Penerapan Data Mining Dalam Prediksi Penjualan Prabot Rumah Tangga Menggunakan Metode Apriori Pada Toko Hasanah Mart. Dalam penelitian ini dijelaskan bahwa penelitian ini melakukan penelitian kuantitatif berupa penelitian yang menggunakan model sistematis yang akan menerapkan hasil dari perhitungan menggunakan sebuah pendekatan *Algoritma Apriori* yang dapat dirancang menggunakan sistem komputerisasi maupun aplikasi sistem yang sudah ada, pada penelitian ini juga dilakukan analisa terhadap data penjualan pada toko Hasanah Mart, bertujuan untuk dapat melihat pola pembelian terbanyak bertujuan agar dapat membuat data barang prabotan yang paling banyak di stok agar tetap ada persediaan barang, mengatur kembali kombinasi tata letak produk agar mudah ditemukan sebagai langkah strategi pemuasan konsumen terhadap kebutuhan barang.

Penelitian yang dilakukan oleh Rini (2019) dengan judul Sistem Penunjang Keputusan Untuk Promo *Bundling* Produk Dengan Metode Saw Dan Apriori. Dalam penelitian ini dijelaskan bahwa penelitian ini melakukan aturan asosiasi dengan metode apriori dimana menggunakan data histori transaksi masa lalu mendeteksi kumpulan atribut dalam hal ini produk yang sering muncul secara bersamaan. Pengetahuan tersebut dapat digunakan untuk merancang promo pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu atau *bundling* produk. Penambahan fitur pemilihan produk menggunakan SAW menjadi nilai tambah karena pemilik toko dapat menentukan sendiri produk dengan kriteria apa yang akan dibuatkan *bundling* produk. Dari hasil pengujian sistem yaitu simulasi yang dilakukan oleh pemilik sebuah toko/pengguna menyatakan bahwa sistem rekomendasi ini memudahkan pengguna untuk membuat *bundling* produk sesuai dengan kriteria yang diinginkan.

II.2 Uraian Teoritis

Penelitian yang dilakukan dengan cara langsung pada Implementasi *Algoritma* Apriori Pada Penjualan Ekspor Barang di PT. *Seafood* Sumatera Perkasa, menggunakan beberapa teoritis serta aplikasi atau bahasa pemrograman, yaitu :

II.2.1. Sistem Informasi

Menurut (Rahmat Alfian Syahputra : 2020) sistem informasi merupakan sistem yang menyediakan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerima informasi. Seiring dengan perkembangan teknologi

yang begitu pesat, sistem informasi yang berbasis *web* dimanfaatkan sebagai sarana peningkatan informasi. Pemanfaatan tersebut akan mempermudah suatu pekerjaan seperti halnya pengolahan data lebih cepat, keputusan yang akan diambil lebih tepat, menghemat waktu dan biaya. Selain itu, sistem informasi yang berbasis *web* juga dapat menjadi sarana promosi yang efisien dan sumber informasi yang dapat diakses oleh pengguna *internet* yang semakin lama semakin luas.

II.2.2. Metode Apriori

Algoritma apriori atau *Association Rule* merupakan salah satu dari *Algoritma* Data Mining yang dirancang agar sistem komputer dapat menggunakan aturan-aturan asosiasi dengan cara mencari pola antara satu *item* dengan *item* lainnya atau hubungan satu *item* atau lebih dari satu *item* dalam suatu data. *Algoritma* ini paling banyak digunakan dalam sistem dan strategi penjualan. (Natalia : 2020).

Algoritma apriori adalah *Association Rule* atau *Algoritma* apriori merupakan salah satu *algoritma* yang pada dasarnya untuk menemukan *Frequentitemsets* (kelompok setiap *item-item* yang memenuhi syarat minimum *support*,) dari setiap data transaksi. (Nina Agustiani : 2020).

Menurut (Rini : 2019) Apriori adalah salah satu dari *algoritma* aturan asosiasi dalam data mining. Asosiasi yaitu mendeteksi kumpulan atribut yang sering muncul secara bersamaan dan membentuk sejumlah kaidah dari kumpulan

tersebut. Hal ini akan menghasilkan aturan ketergantungan yang dapat memprediksi kejadian dari satu item berdasarkan kejadian *item* lainnya. Sebuah aturan yang menyatakan hubungan antara beberapa atribut sering disebut *afinitas* atau analisis market basket. Analisis asosiasi juga dikenal sebagai salah satu teknik data mining yang menjadi dasar dari banyak teknik data mining lainnya. Salah satu tahap analisis asosiasi, yang menarik banyak peneliti untuk menghasilkan *algoritma* yang efisien adalah analisis pola frekuensi tinggi (Pertambahan pola sering). *Algoritma* Apriori digunakan untuk mencari pasangan produk yang dimungkinkan dapat laris terjual jika kedua barang tersebut dibundling.

$$Support(A, B) = \frac{\sum_{k=0}^n \binom{n}{k} \text{Amount of transactions containing } A \text{ and } B}{\sum_{k=0}^n \binom{n}{k} \text{Transaction}}$$

$$Confidence(A, B) = \frac{\sum_{k=0}^n \binom{n}{k} \text{Amount of transactions containing } A \text{ and } B}{\sum_{k=0}^n \binom{n}{k} \text{Amount of antecedent}}$$

Untuk mengukur kekuatan aturan asosiasi ini, digunakan ukuran support dan *confidence*. *Support* adalah *rasio* antara jumlah transaksi yang memuat *antecedent* dan *consequent* dengan jumlah transaksi. *Confidence* adalah *rasio* antara jumlah transaksi yang meliputi semua *item* dalam *antecedent*.

II.2.3. Laravel

Menurut (Dinni Ambriani : 2020) Laravel adalah *framework* berbasis PHP yang sifatnya *open source*, dan menggunakan konsep *model – view – controller*. Laravel berada di bawah lisensi MIT *License* dengan menggunakan *Github* sebagai tempat berbagi *code*. Pada Desember 2013, *laravel* sebagai *framework* terpopuler. Dalam penggunaannya *laravel* memiliki beberapa kekurangan salah satunya yaitu ukuran file yang cukup besar. Di dalam *laravel* terdapat file yang sifatnya *default* seperti *vendor*. File tersebut tidak boleh dihapus sembarangan sehingga ukuran *website* yang dibuat berukuran cukup besar. Selain itu, dibutuhkan koneksi *internet* untuk *instalasi* dan mengunduh *library* *laravel*, dan PHP minimal versi 5.4 untuk menjalankannya.

II.2.4. Pengertian PHP (*Hypertext Preprocessor*)

Menurut (Fauzi Maulana Kusuma : 2019) PHP adalah singkatan dari PHP *Hypertext Preprocessor*, bahasa *interpreter* yang mempunyai kemiripan dengan bahasa C dan *Perl* yang mempunyai kesederhanaan dalam perintah, yang digunakan untuk pembuatan aplikasi *web*. PHP/F1 merupakan nama awal dari PHP (*Personal Home Page/Form Interface*). Dibuat pertama kali oleh Rasmus Lerdorf. PHP awalnya merupakan program CGI yang dikhususkan untuk menerima *input* melalui *form* yang ditampilkan dalam *browser web*. Dengan menggunakan PHP maka *maintenance* suatu situs *web* menjadi lebih mudah. Proses *update* dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP. PHP merupakan *script* untuk pemrograman *script web*

server-side, script yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.

II.2.5. Pengertian MySQL

Menurut (Fauzi Maulana Kusuma : 2019) MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya.

II.2.6 Basis Data

Secara umum untuk menjelaskan tentang pengertian basis data dapat ditinjau dari dua sisi, pengertian secara kharfiah dan pengertian secara istilah. Menurut pengertian secara kharfiah, basis data terdiri dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai suatu markas atau gudang, tempat bersarang

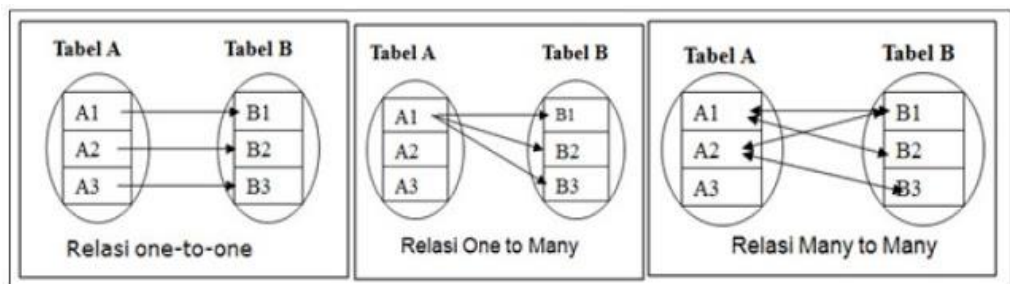
atau tempat berkumpul. Data dapat diartikan merupakan representasi dari fakta dunia yang mewakili suatu obyek (manusia, barang, peristiwa, keadaan dsb) yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. Adapun menurut pengertian secara istilah, terdapat beberapa definisi yaitu sebagai berikut:

1. Himpunan kelompok data (*arsip*) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file/tabel/arsip* yang saling berhubungan yang disimpan dalam media penyimpanan tertentu.
4. Kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi.

II.2.7 Relasi

Relasi menyatakan hubungan antara dua atau beberapa entitas. Setiap relasi mempunyai batasan (*constraint*) terhadap kemungkinan kombinasi entitas yang berpartisipasi. Batasan tersebut ditentukan dari situasi yang diwakili relasi tersebut. Ragam atau jenis relasi dibedakan menjadi beberapa macam antara lain adalah:

1. Relasi *Binary*. Relasi binary merupakan relasi antara dua entitas. Relasi *binary* ini dibedakan menjadi :
 1. Relasi *One-to-one* (notasi 1:1)
 2. Relasi *One-to-many* (notasi 1:N) atau *many-to-one* (notasi N:1)
 3. Relasi *Many-to-many* (notasi M:N)[6]
2. Relasi *Ternary*. Relasi *ternary* adalah merupakan relasi antara tiga entitas atau lebih.
 1. Dalam Relasi *One-to-one* (1:1) setiap atribut dari satu entitas berpasangan dengan satu *attribute* dari entitas yang direlasikan.
 2. Dalam relasi *One-to-many* (1:N) atau *many-to-one* (N:1) satu *attribute* berelasi dengan beberapa *attribute* dari entitas yang direlasikan.
 3. Dalam *Many-to-many* (M:N) satu *attribute* berelasi dengan beberapa *attribute* dari entitas yang direlasikan. Begitu pula sebaliknya.



Gambar II.1 Ragam relasi antar entitas

(Sumber : Abdul Munif, 43)

Sebagaimana entitas dalam relasi juga dapat dibedakan menjadi relasi kuat dan relasi lemah. gambar dibawah ini menjelaskan notasi umum untuk relasi kuat dan relasi lemah.



Gambar II.2 Notasi relasi entitas

(Sumber : Abdul Munif, 43)

II.2.8 Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan atau mendekomposisi atau memecah data menggunakan cara–cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan–penyimpangan (*anomalies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan inefisiensi pengolahan. Proses normalisasi akan menghasilkan relasi yang optimal, yaitu:

1. Memiliki struktur *record* yang mudah untuk dimengerti.
2. Memiliki struktur *record* yang sederhana dalam pemeliharaan.
3. Memiliki struktur *record* yang mudah untuk ditampilkan kembali untuk memenuhi kebutuhan pemakai.
4. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

II.2.9 UML (*Unified Modelling Language*)

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.

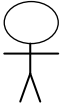
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

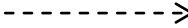
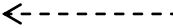
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

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. Simbol-simbol yang digunakan dalam *use case diagram*, yaitu :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa

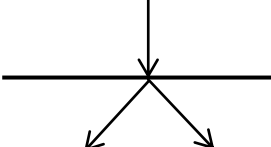
	peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

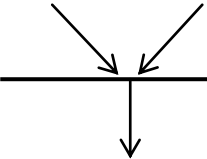
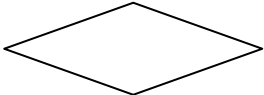
(Sumber : Windu Gatta : 4-6)

2. Diagram Aktivitas (*Activity Diagram*)

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

Tabel II.2. Simbol *Activity Diagram*

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

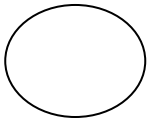
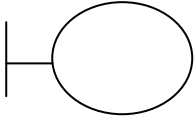
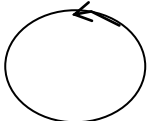
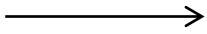
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

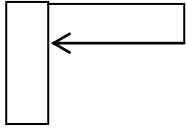
(Sumber : Windu Gatta : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .

	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gatta : 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan 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.4. *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 Gatta : 8-9)