BAB II

LANDASAN TEORI

2.1 Algoritma Apriori

Algoritma apriori adalah salah satu algoritma pengambilan data dengan aturan asosiasi (*Association rule*) untuk menentukan hubungan asosiasi suatu kombinasi item (Kusrini, 2009). Aturan yang menyatakan asosiasi terhadap beberapa atribut seringkali disebut *market basket analysis* atau *affinity analysis*.

Association rules adalah merupakan suatu proses pada data mining untuk menentukan semua aturan asosiasi antara suatu kombinasi item. *Association rules* memiliki dua parameter yaitu syarat minimum *support* (*minimum support*) dan *confidence* (*minimum confidence*) pada sebuah *database* untuk mengetahui penting tidaknya suatu aturan asosiasi. *Support* adalah nilai persentase kombinasi sebuah item dalam database dan *confidence* adalah kuatnya hubungan keterkaitan antar item dalam database. *Database* adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik (Achmad, 2016).

Kedua syarat tersebut akan digunakan untuk *association rules* dengan dibandingkan dengan batasan minimum *support* dan minimum *confidence* yang telah ditentukan. Dengan menggunakan pola asosiasi dapat memberikan gambaran terhadap karakteristik yang sering muncul dalam proses transaksi.

Gambaran aturan asosiasi dinyatakan dalam bentuk berikut :

{ susu, roti } →{ gula } (support = 30%, confidence = 50%)

Pernyataan tersebut dapat diartikan bahwa 50% transaksi yang ada di database memuat item susu dan roti juga memuat gula. Serta 30% dari seluruh transaksi di database memuat ketiga item tersebut.

Metode dasar analisis asosiasi dapat dijabarkan dengan dua tahap sebagai berikut :

1. Penentuan analisa pola frekuensi tinggi (*frequency itemset)*

Pada tahap ini melakukan pencarian kombinasi item yang memenuhi syarat minimum dari nilai *support* yang telah ditentukan dalam database. Untuk mendapatkan nilai *support* item dapat diperoleh dengan rumus sebagai berikut :

$$Support\ (A) = \frac{\Sigma \text{Transaksi item A}}{\Sigma \text{ Transaksi}} \quad \text{.....(1)}$$

Meurut **Larose** (2014), cara untuk menemukan *support* dari dua item yaitu item A dan item B digunakan rumus berikut :

$$Support\ (A \cap B) = \frac{\Sigma \text{Transaksi item A} \cap \text{B}}{\Sigma \text{ Transaksi}} \quad \text{.....(2)}$$

2. Pembentukan Aturan Asosiasi (*Rule Generation)*

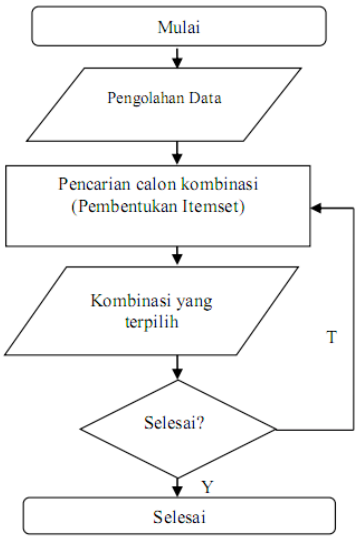
Setelah ditemukan pola frekuensi tinggi maka langkah selanjutnya adalah menemukan aturan asosiatif yang memenuhi syarat *minimum confidence* yang merupakan ukuran ketepatan suatu *rule* yang terkandung dalam Item A dan Item B dapat menggunakan rumus sebagai berikut :

$$Confidence\ (B|A) = \frac{\Sigma \text{Transaksi item A} \cap \text{B}}{\Sigma \text{ Transaksi A}} \quad \text{.....(3)}$$

Hasil dari tahap ini akan ditemukan pola frekuensi tinggi yang sesuai dengan nilai *minimum support* dan *minimum confidence.*

Algoritma apriori dianggap algoritma yang paling stabil dalam menemukan pola frekuensi tinggi dibandingkan dengan pengembangan algoritma lainnya seperti *FP-Growth, dan Hash Based.* Algoritma Apriori dibagi menjadi beberapa tahap yang disebut iterasi atau pass.

Cara kerja atau tahapan kerja dari Algoritma Apriori dapat digambarkan sebagai berikut :



Gambar 2. 1 *Flowchart* Algoritma Apriori

Tiap iterasi menghasilkan pola frekuensi tinggi dengan panjang yang sama dimulai dari iterasi pertama yang menghasilkan pola frekuensi tinggi dengan panjang satu. Untuk setiap iterasi ke-*k* dibagi menjadi beberapa bagian tahapan :

a. Pembentukan kandidat itemset, Kandidat *k-itemset* dibentuk dari kombinasi *(*k-1*)-itemset* yang didapat dari iterasi sebelumnya. Satu ciri dari Algoritma apriori adalah adanya pemangkasan kandidat *k-itemset* yang merupakan himpunan bagian yang berisi *k*-1 item tidak termasuk dalam pola frekuensi tinggi dengan panjang *k*-1.

b. Penghitungan *support* dari tiap kandidat *k-itemset*. *Support* dari tiap kandidat *k-itemset* diperoleh dengan men-*scan* database untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat *k-itemset* tersebut. Ini adalah juga ciri dari Algoritma apriori dimana diperlukan penghitungan dengan *scan* seluruh database sebanyak *k-itemset* terpanjang.

c. Tetapkan pola frekuensi tinggi. Pola frekuensi tinggi yang memuat *k* item atau *k-itemset* ditetapkan dari kandidat *k-itemset* yang memiliki *support* lebih

besar dari *minimum support* yang telah ditentukan.

d. Bila tidak diperoleh pola frekuensi tinggi baru maka seluruh proses dihentikan dan kembali ke tahapan awal

2.2 Sistem Rekomendasi

Sistem rekomendasi adalah sebuah suatu alat dan teknik yang menyediakan saran terkait suatu hal untuk dapat dimanfaatkan oleh user (Ricci, 2011). Pada algoritma apriori rekomendasi adalah menemukan aturan asosiasi keterkaitan antar item sehingga diperoleh pola keterkaitan item (Novriansyah, 2014).

Pada layanan transaksi jual beli terdapat sistem rekomendasi terkait dengan pola pembelian pengguna tersebut pada data sebelumnya, dimana pola pembelian tersebut dapat dijadikan acuan dalam memberikan rekomendasi barang untuk referensi pembelian stok agar meminimalisir kerugian.

2.3 *Hypertext Preprocessor* (PHP)

PHP merupakan salah satu bahasa yang digunakan dalam pembuatan website. PHP diproses pada server web dan hasil output ditampilkan dalam web browser. PHP merupakan salah satu bahasa pemrograman yang scripting yang digunakan dalam pembuatan website (Achmad, 2016).

PHP diproses pada server web dan hasil output ditampilkan dalam web browser yang kemudian diakses oleh *client* melalui laptop, komputer, handphone, dan sebagainya.

2.4 MySQL

MySQL pertama kali dirintis oleh seorang programmer bernama Michael Widenius. MySQL merupakan suatu database server yang cukup populer

disebabkan MySQL menggunakan SQL sebagai bahasa dalam mengakses databasenya dan kelebihan lain ialah MySQL bersifat gratis sehingga seseorang dapat dengan mudah untuk mendapatkan, menggunakan, dan mengembangkannya. Sebuah database mengandung satu dan sejumlah table, terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom (Yanto, 2016).

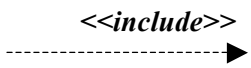
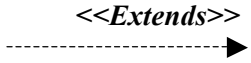
2.5 PHPStorm

Aplikasi text editor yang digunakan untuk membuka file apapun namun sejatinya para programmer menggunakannya untuk menulis code. PHPStorm merupakan salah satu text editor yang sangat ringan, tidak menggunakan banyak memori, dan didukung oleh banyak sekali plugin. Keunggulan lain dari PHPStorm ialah *Autosuggest* / *Intelisense* yang super responsive baik untuk PHP, HTML, CSS, dan Jquery, Terdapat tanda penutup *tag* dan pasangan serta terdapat akses langsung ke dalam GIT sehingga mempermudah tidak lagi menggunakan *Command Prompt* (Ankur Kumar, 2014).

2.6 UML

UML (*Unified Modelling Languange*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan - permasalahan yang kompleks sedemikian rupa sehingga akan lebih mudah dipelajari dan dipahami (Nugroho, 2010).

Tabel 3. 1 Keterangan *Attribute* UML

Notasi / Relasi	Nama	Keterangan
	*Actor*	Untuk menggambarakan seseorang atau siapa saja yang berhubungan dengan sistem
	*Use Case*	Untuk menggambarkan bagaimana seorang *actor* menggunakan sistem
	Relasi Asosiasi	Untuk menggambarkan hubungan antara *actor* dan *use case*
	Relasi *include*	Digunakan jika satu *use case* menggunakan fungsionalitas yang disediakan *use case* lain
	Relasi *extends*	Digunakan jika satu *use case* menggunakan fungsionalitas yang disediakan *use case* lain

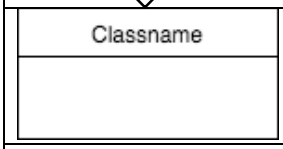
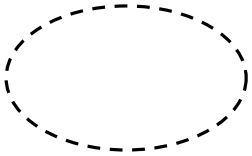
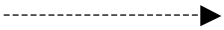
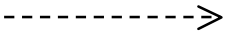
Analisa perancangan sistem dalam penelitian ini menggunakan UML (*Unified Modeling Language*) diantaranya :

a. *Use Case* Diagram

Use Case diagram jenis diagram pada UML (*Unified Modeling Language)* yang menggambarkan interaksi antara sistem dan aktor, *Use Case* diagram juga dapat mendeskripsikan tipe interaksi antara pengguna sistem dengan sistem. (Wiwin, 2015) Dalam penelitian ini, pengguna sistem tersebut adalah admin.

Tabel 3. 2 Keterangan Attribut Use Case Diagram

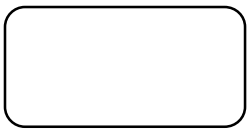
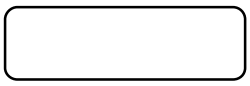
Notasi / Relasi	Nama	Keterangan
	Generalization	Hubungan dimana objek anak (*descendant)* berbagi perilaku dan struktur data dari objek

Notasi / Relasi	Nama	Keterangan
		yang ada diatasnya (*ancestor*)
	*Nary Association*	Upaya untuk menghindari asosiasi lebih dari 2 objek
	*Class*	Himpunan dari objek-objek yang berbagi *attribute* serta operasi yang sama
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang meghasilkan suatu hasil yang terukur bagi suatu aktor
	*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	Association	Apa yang menghubungkan antara objek satu dengan yang lainnya

b. *Activity* Diagram

Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem.

Tabel 3. 3 Keterangan *Atribut Activity Diagram*

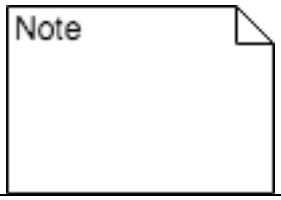
Notasi / Relasi	Nama	Keterangan
	*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	*Action*	*State* dari sistem yang mencermikan eksekusi dari suatu aksi
	*Initial node*	Bagaimana objek dibentuk atau diawali
	*Initiial Final node*	Bagaimana objek dibentuk dan diakhiri/ dihancurkan
	*Fork node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

c. *Sequence* Diagram

Sequence diagram jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan.

Tabel 3. 4 Keterangan *Attribute Sequence Diagram*

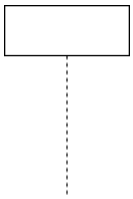
Notasi / Relasi	Nama	Keterangan
	*State*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	*Initial pseudo state*	Bagaimana objek dibentuk atau diawali
	Initial Final state	Bagaimana objek dibentuk dan diakhiri/

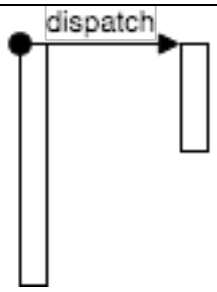
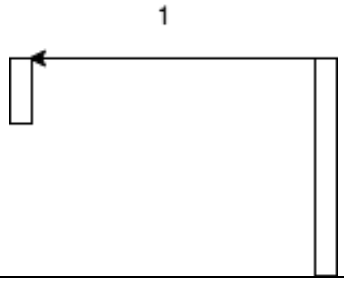
Notasi / Relasi	Nama	Keterangan
		dihancurkan
	*Transition*	Sebuah kejadian yang memicu sebuah *state* objek dengan cara memperbaharui satu atau lebih nilai *attributnya*
	Association	Apa yang menghubungkan antara objek satu dengan yang lainnya
	*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

d. *Class* Diagram

Class diagram yaitu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.

Tabel 3. 5 Keterangan *Atribute Class Diagram*

Notasi / Relasi	Nama	Keterangan
	*Lifeline*	Objek *entity*, antarmuka yang saling berinteraksi

Notasi / Relasi	Nama	Keterangan
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas-aktifitas yang terjadi
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas-aktifitas yang terjadi