BAB II
TINJAUAN PUSTAKA

2.1. Aplikasi *Mobile*

Aplikasi merupakan suatu program yang digunakan oleh seseorang atau instansi untuk menerapkan sistem berbasis komputer. *Mobile* adalah suatu bentuk yang dapat dipindahkan dari satu tempat ketempat yang lain. Sedangkan sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walau dikelola dengan cara berpindah dengan mudah dari tempat satu ketempat yang lain, yang dapat digunakan dengan telpon seluler (Yuhefizar, 2014). Berikut ini adalah karakteristik aplikasi *mobile* adalah sebagai berikut :

1. Ukuran yang kecil : memiliki ukuran yang kecil untuk mempermudah dan membuat nyaman pengguna.
2. *Memory* yang terbatas : perangkat *mobile* memiliki media penyimpanan RAM dan disk yang lebih kecil.
3. Daya proses yang terbatas : memiliki layar yang terbatas tidak seperti rekan mereka yaitu desktop
4. Mengonsumsi daya yang rendah : menghabiskan sedikit daya
5. Kuat dan dapat diandalkan : karena mereka dapat dibawa kemana-mana sehingga harus cukup kuat terkena benturan, geseran, dan tetesan air.
6. Konerktivitas yang terbatas : memiliki bandwith rendah yang tidak langsung tersambung
7. Masa hidup yang pendek : perangkat ini menyala hanya dalam hitungan detik dan kebanyakan mereka selalu menyala.

2.2. Penjualan

Penjualan merupakan bagian dari pemasaran yaitu transaksi jual beli antara pelanggan dan penjual yang menghasilkan keuntungan dari dua belah pihak (Zulkarnain, 2017).

2.3. Data Mining

Data mining adalah suatu proses untuk menemukan hubungan yang berarti seperti pola yang kecendrungan dengan memeriksa dalam kumpulan data yang besar yang menggunakan teknik statistic dan matematika (Larose, 2015).

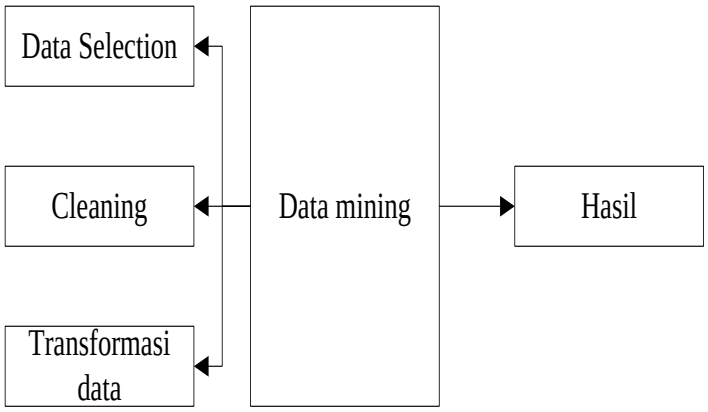
Tugas asosiasi dalam data mining adalah menemukan atribut yang muncul dalam satu waktu yang sering disebut dengan analisis keranjang belanja. Karakteristik *data mining* diantaranya :

a) *Data mining* berhubungan dengan penemuan suatu yang terpsembuji berupa pola data yang tidak diketahui sebelumnya
b) *Data mining* biasanya dapat digunakan dengan data yang sangat besar dan menghasilkan hasil yang lebih terpercaya.
c) *Data mining* berfungsi untuk membuat suatu keputusan yang kritis dalam sebuah strategi

Data mining mmemiliki yang memiliki suatu akar yang panjang dari berbagai bidang ilmu pengetahuan. Beberapa metode yang digunakan dalam data *mining* adalah *vlustering, classification, association rules mining, neural network, genetic* algoritm, dan masih banyak lainnya (Davies and Beynon, 2004).

2.4. Apriori

Apriori merupakan teknik data mining yang menjadi dasar dati berbagai teknik data mining lainnya, teknik ini dangat menarik untuk menghasilkan algoritma yang efisien dengan alanisis pola frekuensi tertinggi (Larose, 2015).



Gambar 2. 1Tahapan Apriori

Berikut adalah penjelasan tahapan apriori preprocessing, yaitu :

a. *Data Selection*

   Tahapan awal yang dilakukan dalam mempersiapkan data, persiapan data yang akan diolah serta melakukan penyeleksian data dibutuhkan untuk menyeleksi data yang benar-benar dibutuhkan pada proses selanjutnya.

b. *Data Cleaning*

   Adalah proses untuk menghilangkan data yang tidak relevan.

   Terkadang hipotesa yang dibuat digunakan untuk membandingkan data yang akan dibuang. Sehingga dapat dengan mudah menemukan hasil pada proses selanjutnya.

c. *Data Transformation*

   Data transformation yaitu mengubah suatu data supaya diperoleh data yang lebih berkualitas. Yang akan dilakukan antara lain menghilangkan noise dari data (smoothing), meng-agregasi data, generalisasi data, normalisasi data, dan pembentukan atribut/fitur.

d. *Datamining*

   Menerapkan teknik data mining untuk menemukan informasi mengenai data transaksi.

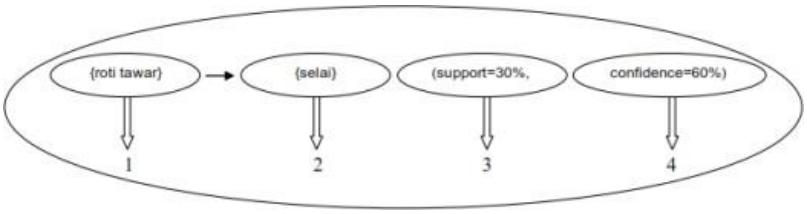
Untuk membentuk kandidat itemset ada dua proses utama yang dilakukan algoritma apriori (Han and Kamber, 2016) :

1. *Join Step* (Penggabungan) Pada proses ini setiap item dikombinasikan dengan item lainnya sampai tidak terbentuk kombinasi lagi.
2. *Prune Step* (Pemangkasan) Pada proses ini, hasil dari item yang dikombinasikan tadi kemudian dipangkas dengan menggunakan minimum support yang telah ditentukan oleh user.

2.5. *Association rules*

Association rule merupakan suatu prosedur yang mencari suatu hubungan berupa relasi antara item satu dengan item lainnya. *associations rule* biasanya menggunakan “if” dan “then”. Dalam menentukan *association rules* perlu

ditentukan *support* dan *confidence* agar dapat membatasi apakah rule tersebut interesting atau tidak (Cormen, 2015).



Gambar 2. 2 Bentuk Umum Aturan Asosiasi

2.5.1. Analisa Pola Frekuensi Tinggi

Pada tahapan ini, melakukan pencarian kombinasi item yang dapat memmenuhi syarat minimum dari nilai *support* dalam suatau *database*. Nilai *support* item diperoleh dengan Persamaan 1.

$$\textit{Support}\ (A) = \frac{\text{Jumlah Transaksi Mengandung A}}{\text{Total Transaksi}} * 100\% \quad \text{.....(1)}$$

Sedangkan Nilai *Support* 2 *Item* diperoleh dari persamaan 2.

$$\textit{Support}\ (A,B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi}} * 100\% \quad \text{.....(2)}$$

2.5.2. Pembentukan Aturan Assosiatif

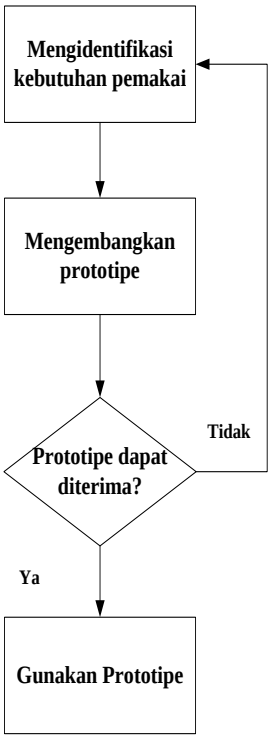
Setalah melakukan semua pola frekuensi tinggi telah ditemukan, maka mencari aturan assosiatif yang memenuhi syarat *minimum* untuk nilai *confidence* dengan mengitung nilai *confidence* dalam sebuah aturan asosiatif. Nilai *confidence* dari aturan diperoleh dari Persamaan 3.

$$\textit{Confidence} = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Jumlah Transaksi Mengandung A}} * 100\% \quad \text{.....(3)}$$

2.6. Metode Pengembangan *Prototype*

Prototype merupakan sebua teknik yang disediakan untuk para pengembang dan calon pengguna yang dapat mengambarkan sistem yang akan dibangun dengan fungsi yang disepakati(McLoad, 2014).

Empat tahapan metode *prototype* yaitu menghasilkan *prototype* secepat mungkin untuk menghasilkan umpan balik dari pengguna untuk di perbaiki lebih cepat.



Gambar 2. 3 Pengembangan Prototype
Sumber:(McLoad, 2014)

Ada empat tahapan dalam pengembangan sistem model *Evolutionary Prototype*, yaitu sebagai berikut:

1. Identifikasi kebutuhan pemakai

   Pengembang mengidentifikasi terhadap pemakai untuk memperoleh suatu gagasan mengenai apa yang dibutuhkan dari sistem yang akan digunakan.
2. Mengembangkan *prototype*

   Pengembang menggunakan satu atau lebih perkakas *prototyping* untuk mengembangkan satu *prototype*.
3. Menentukan apakah *prototype* bisa diterima atau tidak.

   Tahap ini dilakukan oleh pemakai sistem apakah *prototyping* yang sudah dikembangkan bisa diterima atau tidak. Jika sudah sesuai maka langkah empat akan diambil, jika tidak *prototyping* direvisi dengan mengulangi

langkah satu, dua, dan tiga dengan pemahaman yang lebih baik mengenai kebutuhan pemakai.

4. Gunakan *prototype*

   Tahap ini dilakukan oleh pemakai sistem untuk menggunakan sistem yang telah dibangun.

2.7. Perancangan Sistem UML (*Unified Modeling Language*)

Menurut Pressman (2015) *Unified Modeling Language (UML)* adalah bahasa standar untuk menulis perangkat lunak dalam bentuk gambar. *UML* dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sebuah sistem perangkat lunak. Beberapa jenis diagram *UML* antara lain sebagai berikut:

2.7.1. *Class Diagram*

Menurut Pressman (2015) Unsur-unsur utama dari diagram kelas adalah kotak, yang merupakan ikon yang digunakan untuk mewakili kelas dan *interface*. Setiap kotak dibagi menjadi bagian-bagian horisontal. Bagian atas berisi nama kelas. Bagian tengah berisi daftar atribut kelas dan bagian bawah merupakan *operation* dari kelas tersebut menggambarkan simbol-simbol yang ada pada diagram kelas pada tabel *class* diagram 2.1 di bawah ini:

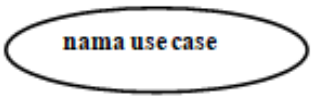
Tabel 2.1 Simbol Class Diagram

Simbol	Deskripsi
Kelas **nama_kelas** +atribut +operasi()	Kelas pada struktur sistem
Antarmuka/*Interface* **nama_*interface***	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi/*asociation*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah/*directed association*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan *multiplicity*
Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)
Kebergantungan/*dependecy*	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/*agregation*	Relasi antar kelas dengan makna semua bagian (*whole-part*)

2.7.2. *Use Case Diagram*

Menurut Pressman (2015), *use case* diagram membantu anda menentukan fungsi dan fitur dari perangkat lunak. Dalam diagram ini, gambar yang menyerupai boneka kayu mewakili aktor yang berhubungan dengan kategori dari pengguna. Di dalam diagram *use case*, para aktor terhubung oleh garis ke *use case* yang mereka kerjakan. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat menggambarkan simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel 2.2 di bawah ini:

Tabel 2.2 Simbol *Diagram Use Case*

Simbol	**Deskripsi**
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*.
Aktor/*actor* 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi/*association*	Komunikasi antara aktor dan *use case* yang berpatisipasi pada *use case* atau *use case* memiliki interaksi dengan actor.
Ekstensi/*extend*	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan .
<<extend>> 	dapat berdiri sendiri walau tanpa *use case* tambahan itu mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek biasanya *use case* tambahan memiliki nama depan.
Generalisasi/*generalization* 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah

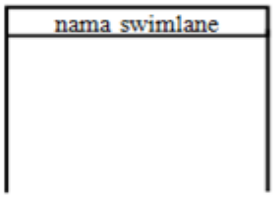
	fungsi yang lebih umum dari lainnya.
Menggunakan/*Include/uses* <<*include*>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini.

2.7.3. *Activity Diagram*

Menurut Pressman (2015), sebuah diagram *activity* menggambarkan perilaku dinamis dari sistem atau bagian dari sistem melalui aliran kontrol antara tindakan yang sistem lakukan. Hal ini mirip dengan sebuah *flowchart* kecuali bahwa suatu diagram *activity* dapat menunjukkan arus bersamaan. Menggambarkan simbol-simbol yang ada pada *activity diagram* dapat dilihat pada Tabel 2.3 di bawah ini :

Tabel 2.3 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.
Percabungan/*decision* 	Asosiasi percabungan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/*join* 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis.
Dokumen 	Menunjukan dokumen sumber atau laporan.

Simbol	Deskripsi
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.8. *jQuery Mobile*

Camden & Matthews (2017) *jQuery Mobile* merupakan *User Interface framework* yang menggunakan *jQuery* sebagai bagian utama untuk pemogramannya. Tidak seperti *framework* lainnya, *jQuery Mobile* fokus pada *HTML* dan *CSS* dengan cara merubahnya menjadi halaman yang *mobile friendly* dan memungkinkan pengguna untuk berinteraksi.

2.9. *MySQL*

MySQL merupakan *software* yang tergolong sebagai *DBMS* (*Database Management System*) yang bersifat *open source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *sourc ecode* (kode yang dipakai untuk membuat *MySQL*) (A.S Rosa & Shalahuddin, 2018).

2.10. XAMPP

XAMPP adalah aplikasi yang berfungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: Apache HTTPServer, MySQLdatabase, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X empat sistem operasi, yang meliputi Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis (Sadeli, 2014).