BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Penelitian Terkait

1) Penelitian terdahulu dengan judul " *Online Shop* kecantikan dan kosmetik dengan pemberian saran pembelian produk menggunakan *Market Basket Analysis*" ,yang dilakukan oleh Anindita Dwi Respita,2015.

   a. Penelitian ini menjelaskan tentang tujuan :

      Merancang dan membuat *online shop* alat kecantikan dan kosmetik dengan pemberian saran pembelian produk menggunakan metode *market basket analysis*.

   b. Aspek Masalah

      Bagaimana menerapkan metode *market basket analysis* menggunakan algoritma apriori sebagai cara perhitungan pola transaksi untuk menentukan rekomendasi produk.

2) Penelitian terdahulu dengan judul "Analisis *Market Basket* dengan Algoritma *Apriori* dan *FP-Growth*", yang dilakukan oleh Erwin,2009.

   a. Aspek Tujuan

      Penerapan *Apriori* dan *FP-growth* dalam proses pencarian *frequent itemset*. Penggunaan *FP-tree* yang digunakan bersamaan dengan algoritma *FP-growth* untuk menentukan

frequent itemset dari sebuah *database*, berbeda dengan paradigma *apriori* yang memerlukan langkah *candidate generation*, yaitu dengan melakukan *scanning database* secara berulang-ulang untuk menentukan *frequent itemset*. Makalah ini juga menyajikan pembahasan mengenai perbandingan kompleksitas waktu antara algoritma *FP-growth* dengan *apriori* dan hasil dari perbandingan algoritma tersebut.

b. Aspek masalah

Algoritma yang umumnya digunakan dalam proses pencarian *itemset* (data yang paling sering muncul) adalah apriori. Tetapi algoritma *apriori* memiliki kekurangan yaitu membutuhkan waktu yang lama dalam proses pencarian itemsets. Untuk mengatasi hal tersebut maka digunakanlah algoritma *FP-growth*.

3) Penelitian terdahulu dengan judul "Sistem Informasi Penjualan pada Grosir *Fashion Online* dengan saran Pembelian Paket Produk menggunakan Algoritma *Apriori*", yang dilakukan oleh Mita Adindayu,2015.

Penelitian ini berusaha untuk menerapkan strategi promosi lain yang masih belum diterapkan oleh GFO yaitu dengan mengelompokkan beberapa produk menjadi satu paket dan menawarkan dengan harga yang lebih rendah, GFO memberikan diskon lebih tinggi jika dalam transaksi pembeli terdapat produk

yang telah ditentukan. Pemaketan produk akan sangat memberikan banyak keuntungan bagi konsumen begitu pula bagi GFO, maka akan lebih baik jika pemaketan produk diterapkan.

Banyaknya transaksi yang masuk akan bermanfaat dalam menentukan paket produk, namun tidak ada kemungkinan jika admin meneliti transaksi secara manual. Untuk itu dibutuhkan aplikasi yang dapat memudahkan pemilihan paket produk. Pada penelitian ini dibangun Sisfo penjualan dengan fitur rekomendasi paket produk yang menggunakan transaksi konsumen sebagai kriterianya, metode yang digunakan yaitu *Market Basket Analysis* dengan algoritma *apriori.* Pada proses penentuan ini dibutuhkan nilai minimum *support* dan minimum *confidence* untuk melihat seberapa sering produk tersebut dibeli dalam satu kali transaksi sebelum aturan asosiasi final ditentukan.

4) Penelitian terdahulu dengan judul "IMPLEMENTASI METODE *MARKET BASKET ANALYSIS (MBA)* PADA APLIKASI *E-COMMERCE* STUDI KASUS ANANDA *SHOP",* yang dilakukan oleh Rita Saraswati,2016.

   a. Aspek Tujuan

      Tujuan dari penelitian ini adalah mengimplementasikan aplikasi *e-commerce* dengan menggunakan metode *Market*

Basket Analysis (MBA) yang dapat mempermudah dan memperluas pemasaran pada toko Ananda.

b. Aspek Masalah

masalah dalam penelitian ini adalah bagaimana merancang aplikasi *E-Commerce* toko Ananda dengan menggunakan metode *Market Basket Analysis* (MBA) yang dapat memperluas dan mempermudah penjualan serta dapat memenuhi persaingan dalam pemasaran.

2.2 Landasan Teori

2.2.1 Sistem informasi

Sistem informasi adalah suatu cara yang terorganisir mengumpulkan, memasukan dan memproses data, mengendalikan, dan menghasilkan informasi dengan berbasis proses manual atau Komputer untuk mencapai sasaran dan tujuan organisasi (Anisah,2007).

2.2.2 Market Basket Analisis (MBA)

Market basket analysis adalah suatu metodologi untuk melakukan analisis *buying habit* konsumen dengan menemukan asosiasi antar beberapa item yang berbeda, yang diletakkan konsumen dalam *shopping basket* yang dibeli pada suatu transaksi tertentu. Tujuan dari *market basket* adalah untuk mengetahui produk-produk mana yang mungkin akan dibeli secara bersamaan. Analisis data transaksi dapat menghasilkan pola pembelian produk

yang sering terjadi. Teknik ini telah banyak digunakan oleh toko grosir maupun retail.

Market Basket Analysis memanfaatkan data transaksi penjualan untuk dianalisis kemudian menemukan pola item-item yang secara bersamaan dalam suatu transaksi. Salah satu manfaat dari *Market Basket Analysis* merancang strategi penjualan atau pemasaran dengan memanfaatkan data penjualan yang ada di perusahan yaitu : (Respita, 2015)

a) Dengan mengubah tata letak toko, menempatkan item-item barang secara berdekatan yang sering di beli secara bersamaan oleh konsumen.
b) Memberikan diskon kepada item barang yang jarang di beli dan mahal.

Market basket Analysis merupakan salah satu contoh penerapan Association Rule. Fungsi *Association Rules* sering kali disebut dengan *"market basket analysis"*, yang digunakan untuk menemukan relasi atau korelasi diantara himpunan item. *Market Basket Analysis* adalah Analisis terhadap kebiasaan membeli *customer* dengan mencari asosiasi dan korelasi antara item-item berbeda yang diletakkan *customer* dalam keranjang belanjaannya. Fungsi ini paling banyak digunakan untuk menganalisa data dalam rangka keperluan strategi pemasaran, desain katalog, dan proses pembuatan keputusan bisnis. Tipe *association rule* bisa dinyatakan sebagai misal : "80% dari orang-orang yang membeli roti, mie dan

saus akan membeli juga susu". Aturan asosiasi *mengcapture item* atau kejadian dalam data berukuran besar yang berisi data transaksi. Dengan kemajuan teknologi, data penjualan dapat disimpan dalam jumlah besar yang disebut dengan "basket data." Aturan asosiasi yang didefinisikan pada basket data, digunakan untuk keperluan promosi, desain katalog, segmentasi *customer* dan target pemasaran. Secara tradisional, aturan asosiasi digunakan untuk menemukan *trend* bisnis dengan menganalisa transaksi *customer* (Amiruddin *dkk*, 2008)

a. Prinsip Kerja Metode *Market Basket Analysis*

Metode *market basket analysis* atau disebut juga dengan metode *association rule* memiliki dua tahapan untuk menemukan informasi *(knowledge*) dari suatu *database*.

Tahapan pertama adalah menemukan *frequent itemset* dan tahapan kedua adalah membentuk aturan assosiasi berdasarkan *frequent itemset*.

Langkah yang akan dilakukan adalah sebagai berikut.

1. Mencari *frequent itemset* Pada tahap ini mencari kombinasi item yang memenuhi syarat *support* dalam database transaksi. Rumus perhitungan *support* adalah sebagai berikut :

$$Support\ (A) = \frac{jumlahTransaksiMengandungA}{jumlah\ transaksi} \times 100$$

2. Membentuk aturan assosiasi Setelah semua *frequent itemset* telah diketahui, selanjutnya adalah membentuk aturan assosiasi yang memenuhi nilai *confidence*. Untuk mengetahui nilai *confidence* perhitungannya adalah sebagai berikut :

$$\text{Confidence : } P(A \cap B) = \frac{\sum TransaksiMengandungAdanB}{\sum TransaksiMengandungA} \text{ X } 100\%$$

2.2.3 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.* (Erwin, 2009).

Penggalian itemset yang *frequent* dengan menggunakan algoritma *FP-Growth* akan dilakukan dengan cara membangkitkan

struktur data *tree* atau disebut dengan *FPTree*. Metode *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu sebagai :

a) 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.

b) 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*.

c) Tahap pencarian *frequent itemset*.

Apabila *conditional FP-Tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi *itemset* untuk setiap conditional *FP-Tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-growth* secara rekursif (proses memanggil dirinya sendiri).

Ketiga tahap tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*, yang dapat dilihat pada gambar 2.1

```
Input : FP-Tree Tree
Output : Rt sekumpulan lengkap pola
frequent
Method : FP-growth (Tree, null)
Procedure : FP-growth (Tree, α)
{
01: if Tree mengandung single path P;
02: then untuk tiap kombinasi (dinotasikan β)
dari node-node dalam path do
03: bangkitkan pola β α dengan support dari
node-node dalam β;
04: else untuk tiap a1 dalam header dari Tree
do
{
05: bangkitkan pola
06: bangun β = a1 α dengan support = a1.
support
07: if Tree β = θ
08: then panggil FP-growth (Tree, β)
}
}
```

Gambar 2.1 algoritma *FP-Growth*

2.2.4 UML

UML(*Unifed Modeling Language*) adalah salah standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

2.2.5 Sejarah Singkat UML

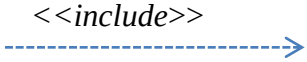
Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada developer pengembang bahasa pemrograman berorientasi objek selanjutnya. Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan

berorientasi objek dan pada bulan september 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan konstribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait didalamnya. Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG (M.Shalahuddin dan Rosa A.S, 2011).

2.2.6 Use Case Diagram

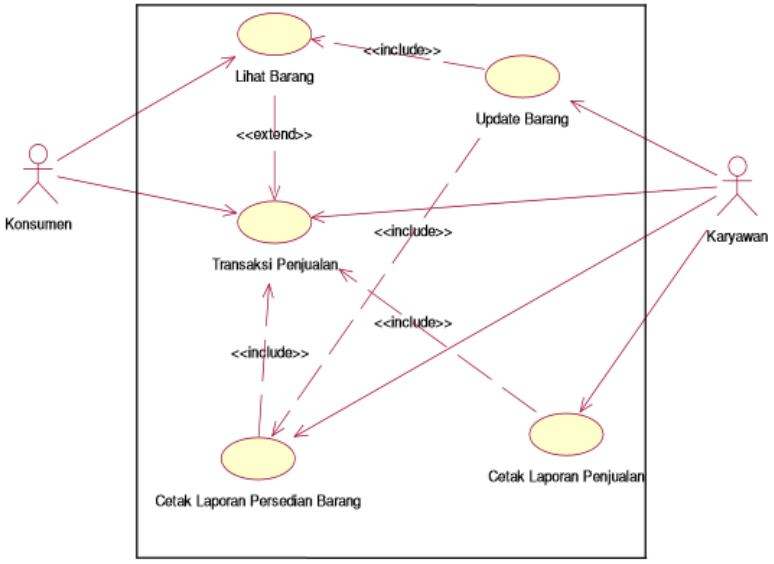
Use case diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2.1 Notasi *use case*

Simbol	**Deskripsi**
	Fungsionalitas yang disediakan *system* sebagai unit-unit yang saling bertukar pesan antarunit atau *actor*.
Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, diluar sistem informasi yang akan dibuat itu sendiri.
Asosiasi/*association*	Komukasi antara *actor* dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan *actor*.
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

Simbol	Deskripsi
Generalisasi/*generalization*	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. *(arah panah biasanya mengarah pada use case yang menjadi generalisasinya/umum)*
Ekstensi/ *extend* <<*extend*>>	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu
Collaboration	Urutan aksi-aksi yang ditampilakan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
Boundary	Digunakan untuk menggambarkan sebuah form
Control class	Untuk menghubungkan *boundary* dengan tabel
Entity class	Untuk menggambarkan hubungan kegiatan yang akan dilakukan
Message( )	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

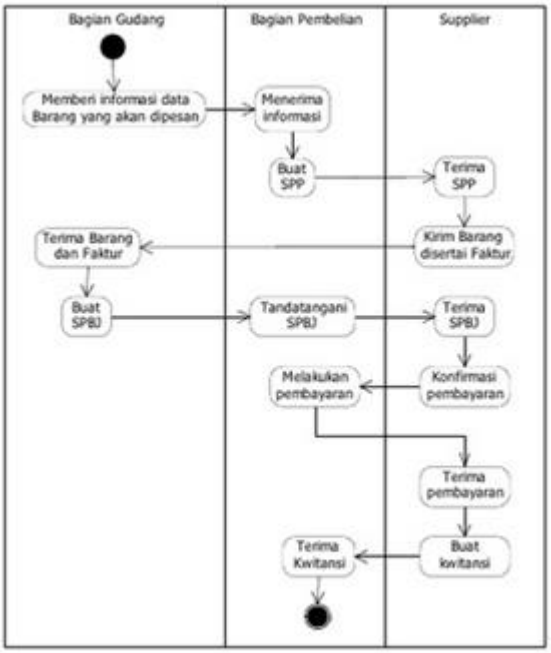
Berdasarkan penjelasan diatas terdapat contoh yang dapat dilihat pada gambar 2.2.



Gambar 2.2 Contoh *Use Case* Diagram

2.2.7 *Activity* Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa activity diagram menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Dapat dilihat dari contoh gambar 2.3

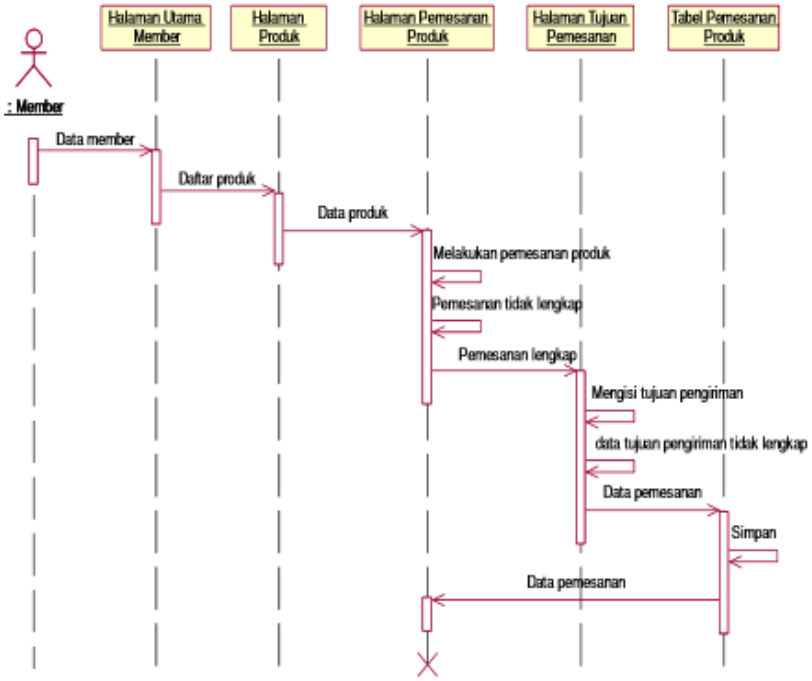


Gambar 2.3 Contoh gambar *Activity* Diagram

2.2.8 *Sequence* Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode

yang dimiliki kelas yang di instansiasi menjadi objek itu. Berdasarkan penjelasan diatas dapat dilihat contoh pada gambar 2.4.



Gambar 2.4 Contoh *Sequence* Diagram

2.2.9 Use Case Realization

Fungsionalitas *use case* direpresentasikan dengan aliran peristiwa-peristiwa. Skenario digunakan untuk menggambarkan bagaimana *use case-use case* direalisasikan sebagai interaksi antara objek-objek. *Use case realization* menggambarkan bagaimana realisasi dari setiap *use case* yang ada pada *use case* model. Untuk menggambarkan bagaimana realisasi dari suatu *use case* dapat menggunakan beberapa diagram diantaranya *class* diagram serta *interaction* diagram.