

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Data Mining**

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam database (Hartono and Nursikuwagus 2016). Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar.

#### **2.2 Tinjauan Penelitian Terkait**

Berikut adalah beberapa penelitian yang terkait dan menjadi referensi pada penelitian ini :

Pada penelitian (Khoiriah and Yanto 2015), membahas tentang analisa data dengan menggunakan data mining dan metode algoritma apriori. Sistem yang dibangun ditujukan untuk pemenuhan dalam penentuan pola pembelian obat dengan menggunakan bahasa pemrograman Visual Basic 6.0 dan database MySQL pada studi kasus di sektor kesehatan. Sistem ini dibangun berdasarkan kebutuhan pengguna yang diperoleh melalui metode wawancara dan studi lapangan. Metodologi pengembangan sistem yang digunakan yaitu metode waterfall yang terdiri analisa, desain, pengkodean, dan pengujian. Hasil pengujian dengan algoritma apriori dan sistem yang dibangun menunjukkan hasil yang telah memenuhi kebutuhan dalam penentuan pola pembelian obat berdasarkan kecenderungan pembelian obat oleh pelanggan. Dibandingkan dengan sistem yang sedang berjalan kinerja tersebut ditujukan pada efektifitas informasi dari sistem tentang penentuan pola pembelian obat untuk ketersediaan obat dan tata letak obat

untuk memudahkan dalam mengetahui keberadaan obat yang dilihat dari 2 itemset obat.

Pada penelitian (Warnilah and Purnia 2017), membahas tentang salah satu optik yang ada di Tasikmalaya, Pada kegiatannya Optik Nasional ini melakukan proses jual beli Kacamata dengan berbagai merk, Banyak nya Transaksi Penjualan sehari-hari maka data penjualan pun semakin lama akan bertambah semakin banyak. Jika dibiarkan saja, maka data-data transaksi penjualan tersebut hanya menjadi sampah yang tidak berarti. Dengan adanya dukungan perkembangan teknologi, semakin berkembang pula kemampuan dalam mengumpulkan dan mengolah data. Pemanfaatan informasi dan pengetahuan yang terkandung di dalam banyaknya data tersebut, pada saat ini disebut dengan data mining. Data-data transaksi tersebut tidak hanya berfungsi sebagai arsip bagi perusahaan, data tersebut dapat dimanfaatkan dan diolah menjadi informasi yang berguna untuk peningkatan penjualan dan promosi produk. Untuk mendapatkan informasi tentang hasil produk yang paling banyak di jual dan diminati di perusaahn optik nasional dari suatu database transaksi, penulis menggunakan algoritma Apriori, sehingga nanti hasilnya dapat digunakan untuk pengembangan peningkatkan penjualan dan pemasaran produk kacamata

### **2.3 Algoritma Apriori**

Algoritma apriori termasuk jenis aturan asosiasi pada data mining. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut affinity analysis atau market basket analysis (Bu'ulolo and Tamba 2019). Analisis asosiasi atau association rule mining adalah teknik data mining untuk menemukan aturan suatu kombinasi item. Salah satu tahap analisis asosiasi yang menarik perhatian banyak

peneliti untuk menghasilkan algoritma yang efisien adalah analisis pola frekuensi tinggi (frequent pattern mining). Penting tidaknya suatu asosiasi dapat diketahui dengan dua tolak ukur, yaitu : support dan confidence. Support (nilai penunjang) adalah persentase kombinasi item tersebut dalam database, seangkan confidence (nilai kepastian) adalah kuatnya hubungan antar-item dalam aturan asosiasi.

Berikut adalah langkah-langkah dalam algoritma apriori :

1. Tentukan minimum support
2. Iterasi 1, hitung item-item dari support(transaksi yang memuat seluruh item) dengan men-scan database untuk 1-itemset, setelah 1-itemset didapatkan, dari 1-itemset apakah diatas minimum support, apabila telah memenuhi minimum support, 1-itemset tersebut akan menjadi pola frequent tinggi.
3. Iterasi 2, untuk mendapatkan 2-itemset, harus dilakukan kombinasi dari k-itemset sebelumnya, kemudian scan database lagi untuk hitung item-item yang memuat support. itemset yang memenuhi minimum support akan dipilih sebagai pola frequent tinggi dari kandidat.
4. Tetapkan nilai k-itemset dari support yang telah memenuhi minimum support dari k-itemset.
5. Lakukan proses untuk iterasi selanjutnya hingga tidak ada lagi k-itemset yang memenuhi minimum support.

Berikut adalah rumus untuk mencari nilai support sebuah item :

$$\text{Support (A)} = \frac{\text{Jumlah transaksi mengandung A}}{\text{Total transaksi}}$$

Shampoo tara untuk mencari nilai support dari 2 item diperoleh dari rumus berikut :

$$\text{Support (A U B)} = \frac{\text{Jumlah transaksi mengandung A dan B}}{\text{Total transaksi}}$$

## 2.4 Algoritma *Moving Average*

*Moving Average* merupakan metode yang sering digunakan untuk memprediksi keadaan di masa depan menggunakan data yang ada di masa lalu (historical data) (Firnando, et al. 2019). Bentuk umum MA diberikan sebagai berikut:

$$\text{MA} = \frac{\sum x}{\text{Jumlah periode}}$$

Dimana :

$\sum x$  = Keseluruhan penjumlahan dari semua data periode waktu yang diperhitungkan.  
Jumlah periode = Jumlah periode rata-rata bergerak.

## 2.5 UML (*Unified Modelling Language*)

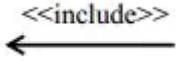
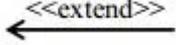
*Unified Modeling Language* (UML) merupakan suatu alat untuk menggambarkan pemodelan sistem (Hendro HS and Arifin 2017). UML merupakan notasi grafis berupa meta-model, yang dapat digunakan untuk menggambarkan dan mendesain sistem perangkat lunak, khususnya sistem pemrograman yang berorientasi objek. Dengan menggunakan UML, pendefinisian masalah dapat dilakukan dengan notasi grafis, sehingga memudahkan dalam pemahaman sistem kompleks. Berikut adalah beberapa diagram yang digunakan dalam *unified modelling language*.

## 1. Use Case Diagram

*Use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif/sudut pandang para pengguna sistem. Use case mendefinisikan “apa” yang dilakukan oleh sistem dan elemen-elemennya, bukan “bagaimana” sistem dan elemen-elemennya saling berinteraksi. Use case bekerja dengan menggunakan “scenario”, yaitu deskripsi urutan-urutan langkah yang menerangkan apa yang dilakukan pengguna terhadap sistem maupun sebaliknya. Use-case diagram mengidentifikasi fungsionalitas yang dimiliki oleh sistem (use case), user yang berinteraksi dengan sistem (actor) dan asosiasi/keterhubungan antara user dengan fungsionalitas sistem. Berikut adalah simbol yang ada pada use case diagram.

**Tabel 2. 1 Use Case Diagram**

Gambar	Keterangan
	Aktor mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan use case.
	Use case, abstraksi dan interaksi antara sistem dan aktor.
	Asosiasi, abstraksi dari penghubung antara aktor dengan use case.
	Generalisasi, menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case.

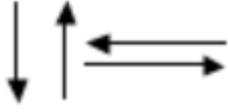
Gambar	Keterangan
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya.
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi.

## 2. Activity Diagram

Diagram activity merupakan diagram aliran kendali antara satu aktifitas ke aktifitas lain. Diagram ini menggambarkan aksi-aksi dan hasil. Diagram activity juga merupakan diagram flowchart yang diperluas. Diagram aktifitas berupa operasi-operasi dan aktifitasaktifitas di use case.

**Tabel 2. 2** Activity Diagram

Gambar	Keterangan
	Memperhatikan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	Bagaimana objek dibentuk atau diawali.
	Bagaimana objek dibentuk dan diakhiri.

Gambar	Keterangan
	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

### 3. Class Diagram

*Class* diagram menggambarkan hubungan antara obyek-obyek yang terlibat didalam sistem, class diagram dapat menunjukkan operasi maupun properti didalam sebuah obyek. Berikut ini adalah class diagram sistem informasi pusat karir yang akan dibangun.

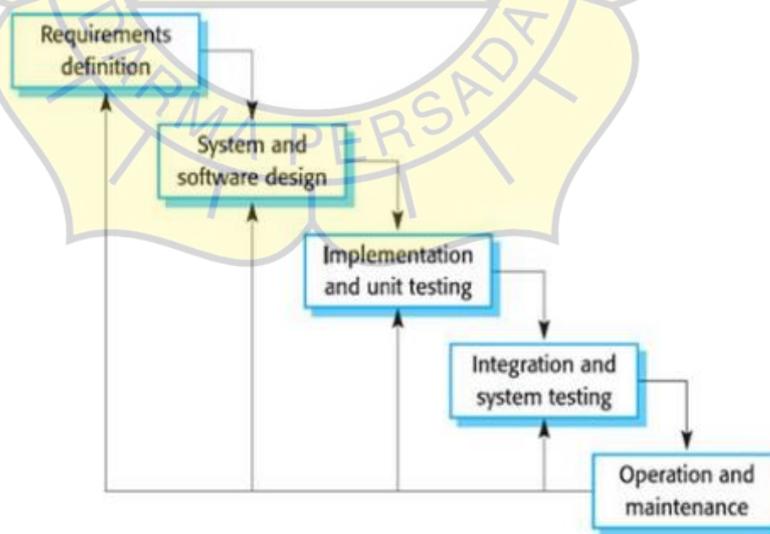
**Tabel 2. 3** Class Diagram

Gambar	Keterangan
	Hubungan, dimana objek anak berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
	Upaya, untuk menghindari asosiasi dengan lebih dari 2 objek.
	Himpunan, dari objek-objek yang berbagi atribut serta operasi yang sama.
	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

Gambar	Keterangan
←-----	Operasi, yang benar-benar dilakukan oleh suatu objek.
----->	Hubungan, dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
-----	Apa yang menghubungkan antara objek satu dengan objek lainnya.

## 2.6 Waterfall

Metode waterfall merupakan model pengembangan sistem informasi yang sistematis dan sekuensial (Sasmito 2017). Metode Waterfall memiliki tahapan-tahapan sebagai berikut :



**Gambar 2. 1** Tahapan Metode Waterfall

### 1. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

### 2. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

### 3. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

## **2.7 *Blackbox Testing***

*Blackbox Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Black Box Testing cenderung untuk menemukan hal-hal berikut :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.