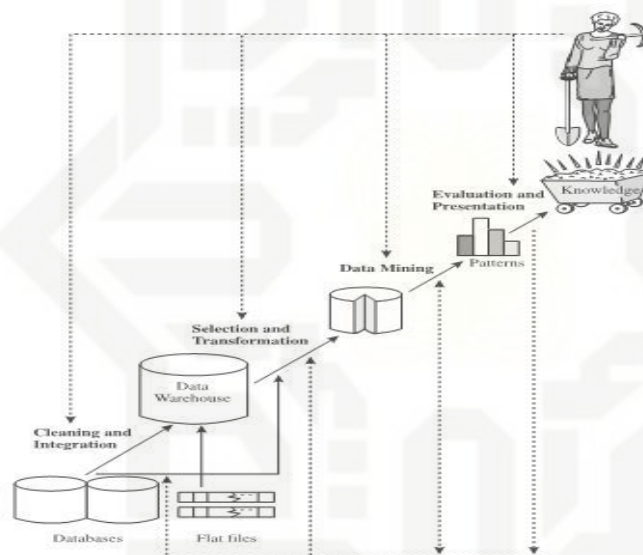


BAB II LANDASAN TEORI

2.1 Knowledge Discovery in Data (KDD)

Istilah *data mining* dan *knowledge discovery in database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain dan salah satu tahapan dalam proses KDD adalah *data mining*. Proses KDD secara garis besar dapat dijelaskan sebagai berikut (Kusrini dkk, 2009)



Gambar 2.1 Tahapan *Knowledge Discovery in Databases*.
 Sumber : (Han, Jiawei, 2012)

Tahap-tahap KDD ada 5 yaitu:

1. *Selection*

Pada tahap ini dilakukan pemilihan atau seleksi data dari sekumpulan data operasional sebelum tahap proses *mining* dimulai. Selanjutnya data yang ada direduksi dan diambil data yang relevan dari *database* untuk dianalisis.

Beberapa cara seleksi data antara lain adalah:

- a. *Dimensionality reduction*, adalah mengurangi jumlah variabel acak atau parameter yang jadi pertimbangan.
- b. *Numerosity reduction*, menggunakan *parametric* atau *non-parametric model* untuk mendapatkan representasi yang lebih kecil dari data asli.
- c. *Data compression*, menerapkan transformasi untuk mengurangi representasi dari data asli.

2. *Preprocessing*

Setelah data selesai diseleksi, data selanjutnya akan dilakukan *preprocessing* terlebih dahulu. Tujuan *preprocessing* adalah agar meningkatkan *performance* dari teknik atau metode *data mining*. Ada beberapa tahapan *preprocessing* sebagai berikut:

I. *Data cleaning*

Data cleaning merupakan proses yang berfungsi untuk membersihkan data yang duplikasi, memeriksa data yang tidak konsisten (inkonsisten). Pada umumnya data yang diperoleh baik dari *database* suatu perusahaan maupun hasil eksperimen, memiliki isi yang tidak sempurna seperti data yang hilang, data yang tidak valid. Berikut beberapa teknik dari proses *data cleaning* sebagai berikut:

- a. *Missing value*
 - a) Mengabaikan record
 - b) Mengisikan secara manual
 - c) Menggunakan nilai *mean/ median* dari parameter yang mengandung *missing value*
 - d) Menggunakan nilai yang paling mendekati.

b. *Noise data*

Noise data adalah suatu kesalahan acak atau variasi dalam variabel terukur.

Beberapa metode noise data dilakukan dengan cara :

- a) *Binning*
 - 1) *Smoothing by bin mean*, semua nilai di tiap bin diganti dengan rata-rata nilai tiap bin.

2) *Smoothing by bin median*, setiap nilai bin diganti dengan nilai yang paling mendekati dari batasan nilai. Batasan nilai terbentuk dari (min, max) dari tiap bin.

II. *Data integration*

Penggabungan data dari berbagai database kedalam suatu *database* baru. Integrasi data dilakukan pada parameter yang mengidentifikasi entitas-entitas yang unik. Ilustrasi dalam *database* seperti *primary key* dan *foreign key*.

3. *Transformation*

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*. Beberapa metode *data mining* membutuhkan format data yang khusus sebelum bisa diaplikasikan. Beberapa cara yang dapat dilakukan dalam transformasi data adalah :

I. *Normalization*, dimana data atribut dibuat dalam skala tertentu sehingga menjadi kisaran data yang lebih kecil sehingga sebaran datanya tidak terlalu jauh. Dengan rumus normalisasi:

a. *Min-Max*

Min-Max merupakan metode normalisasi dengan melakukan transformasi linier terhadap data asli. Rumus:

$$v'_i = \frac{v_i - \min_A}{(\max_A - \min_A)} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A \dots \dots \dots (2.1)$$

Keuntungan dari metode ini adalah keseimbangan nilai perbandingan antar data saat sebelum dan sesudah proses normalisasi. Tidak ada data bias yang dihasilkan oleh metode ini. Kekurangannya adalah jika ada data baru, metode ini akan memungkinkan terjebak pada "out of bound" *error*.

b. *Z-Score*

Z-score merupakan metode normalisasi yang berdasarkan nilai rata-rata (*mean*) dan *standard deviation* (deviasi standar) dari data. Rumus:

$$v'_i = \frac{v_i - \bar{A}}{\sigma_A} \dots \dots \dots (2.2)$$

Metode ini sangat berguna jika kita tidak mengetahui nilai aktual minimum dan maksimum dari data.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

c. *Decimal Scalling*

Metode ini melakukan normalisasi dengan menggerakkan nilai desimal dari data ke arah yang diinginkan. Rumus:

$$v'_i = \frac{v_i}{10^j} \dots\dots\dots (2.3)$$

dimana i adalah nilai integer terkecil untuk menggerakkan nilai ke arah yang diinginkan.

4. *Data mining*

Data mining adalah suatu proses pencarian pola atau informasi penting dan menarik didalam *database* dengan menggunakan metode tertentu. Algoritma yang akan digunakan harus tepat agar tujuan dan proses KDD dapat berjalan dengan baik.

5. *Intepretation*

Intepretation merupakan tahap perubahan pola informasi yang dihasilkan proses data mining dalam bentuk informasi yang mudah dipahami oleh pihak yang terkait, tahapan ini juga mencakup pemeriksaan kesesuaian pola informasi dengan fakta yang ada sebelumnya.

2.2 **Data Mining**

Menurut (Zaki, 2014) *Data mining* adalah proses untuk menemukan suatu pengetahuan, informasi, dan pola baru yang terkandung didalam database yang besar untuk dijadikan sebagai deskripsi yang berguna dan model prediksi.

Menurut (Kushrini dkk, 2009) *Data mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam database. *Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang berkaitan dari berbagai *database* besar.

Menurut (Fatihatul dkk, 2013) *Data Mining* adalah sebuah teknologi baru yang memiliki potensi sangat besar dalam penggalian informasi yang tersembunyi. Data mining juga sering didefinisikan sebagai proses ekstraksi informasi prediktif tersembunyi dari *database* yang sangat besar. Teknologi ini memungkinkan sebuah perusahaan untuk lebih proaktif dalam penyusunan strategi yang efektif dan juga dalam pengambilan keputusan berdasarkan perilaku dan tren masa depan yang dapat diprediksi oleh proses data mining.

Data mining juga mempunyai beberapa karakteristik tertentu. Berikut karakteristik dari *data mining* :

1. *Data mining* berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
2. *Data mining* bisa menggunakan data yang sangat besar. Biasanya data yang besar digunakan untuk membuat hasil yang lebih dipercaya,
3. *Data mining* hanya berguna untuk membuat keputusan kritis, terutama dalam strategi.

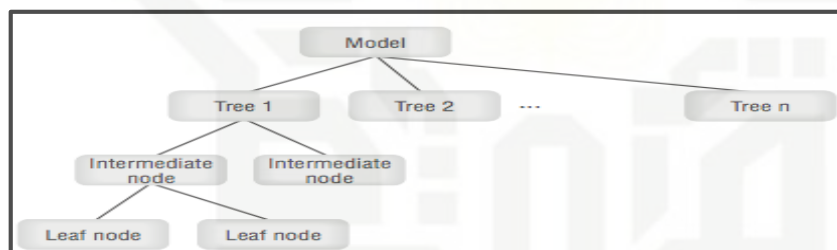
2.1 Fungsi Data Mining

Banyak fungsi *data mining* yang dapat digunakan. Dalam kasus tertentu fungsi data mining dapat digabungkan untuk menjawab masalah yang dihadapi.

Berikut adalah fungsi *data mining* menurut (Maclennan dkk, 2009) :

1. Classification

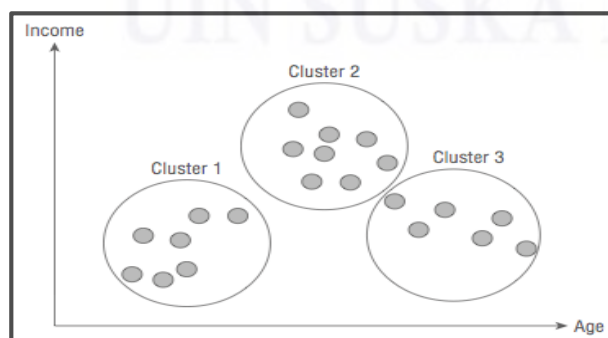
Fungsi dari *Classification* adalah untuk mengklasifikasikan suatu target *class* ke dalam kategori yang dipilih.



Gambar 2.2 Classification - Decision Tree

2. Clustering

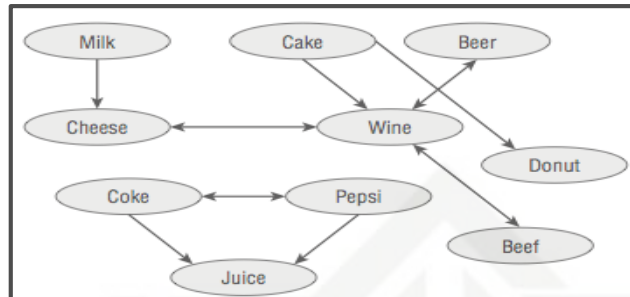
Fungsi dari *clustering* adalah untuk mencari pengelompokan parameter ke dalam segmentasi-segmentasi berdasarkan similaritas.



Gambar 2.3 Clustering

3. Association

Fungsi dari *association* adalah untuk mencari keterkaitan antara parameter atau *item set*, berdasarkan jumlah item yang muncul dan *rule association* yang ada.



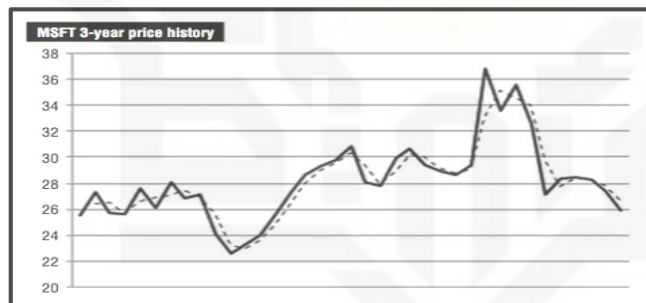
Gambar 2.4 Product Association

4. Regression

Fungsi dari *regression* hampir mirip dengan klasifikasi. Fungsi dari regression adalah bertujuan untuk mencari prediksi dari suatu pola yang ada.

5. Forecasting

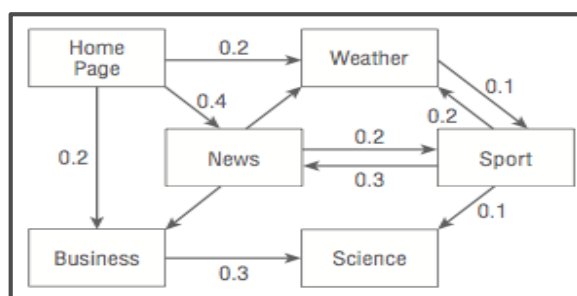
Fungsi dari *forecasting* adalah untuk peramalan waktu yang akan datang berdasarkan *trend* yang telah terjadi di waktu sebelumnya.



Gambar 2.5 Time Series

6. Sequence Analysis

Fungsi dari *sequence analysis* adalah untuk mencari pola urutan dari rangkaian kejadian.



Gambar 2.6 Web Navigation Sequence

7. *Deviation Analysis*

Fungsi dari *deviation analysis* adalah untuk mencari kejadian langka yang sangat berbeda dari keadaan normal (kejadian abnormal).

2.2 Karakteristik Data Mining

Data mining juga mempunyai beberapa karakteristik tertentu. Berikut karakteristik dari *data mining* (Davies & Beynon, 2004):

1. *Data mining* berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
2. *Data mining* bisa menggunakan data yang sangat besar. Biasanya data yang besar digunakan untuk membuat hasil yang lebih dipercaya,
3. *Data mining* hanya berguna untuk membuat keputusan kritis, terutama dalam strategi.

2.3 Association Rules

Association rules merupakan suatu metode untuk menemukan semua aturan asosiatif yang memenuhi syarat minimum untuk support dan syarat minimum untuk confidence pada sebuah database (Fatihatul dkk, 2013).

Dalam menentukan suatu asosiasi rule, terdapat suatu *interestingness measure* (ukuran ketertarikan) yang didapatkan dari hasil pengelolaan data dengan perhitungan tertentu. Terdapat dua ukuran (Jiawei dkk, 2012) yaitu:

1. *Support* (nilai pendukung): suatu ukuran yang menunjukkan seberapa besar tingkat kemunculan suatu item / itemset dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu itemset layak untuk dicari confidence pada tahapan selanjutnya.
2. *Confidence* (nilai kepastian): suatu ukuran yang menunjukkan hubungan antara 2 buah item secara conditional.

Kedua ukuran tersebut akan digunakan dalam menentukan *interesting association rules*, yaitu untuk dibandingkan dengan batasan yang ditentukan oleh user. Batasan tersebut umumnya terdiri dari *support* dan *confidence* yang digunakan pada proses pencarian *association rules*.

Proses pencarian asosiasi terbagi menjadi dua tahap yaitu analisis *frequent itemset* dan pembentukan asosiasi (Jiawei dkk, 2012).

a. Analisis *frequent Item*

Tahapan ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah item diperoleh dengan rumus 2.2 sebagai berikut:

$$Support(x) = \frac{Jumlah\ Transaksi\ yang\ mengandung\ item\ X}{Total\ transaksi} \times 100\% \dots\dots\dots(2.4)$$

Kemudian untuk mencari nilai *support* dari dua item dapat menggunakan rumus berikut:

$$Support(x,y) = \frac{Jumlah\ Transaksi\ yang\ mengandung\ item\ X\ dan\ Y}{Total\ transaksi} \times 100\% \dots\dots\dots(2.5)$$

Setelah semua *frequent item* didapat, maka ditentukan syarat *minimum Confidence*. *Confidence* adalah nilai ukuran seberapa besar valid tidaknya suatu *association rules*. *Confidence* suatu rule R ($x \rightarrow y$) adalah proporsi dari semua transaksi yang mengandung baik x maupun y dengan yang mengandung x.

b. Pembentukan *Association Rule*

Setelah semua pola *frequent Itemset* ditemukan, kemudian dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif $x \rightarrow y$. Nilai *confidence* dari aturan $x \rightarrow y$ diperoleh dari rumus 2.6

$$Confidence(x \rightarrow y) = \frac{Jumlah\ Transaksi\ yang\ mengandung\ item\ X\ dan\ Y}{Total\ transaksi} \times 100\% \dots\dots(2.6)$$

Keterangan :

x dan y = item

Sebuah *association rule* dengan *confidence* sama atau lebih besar dari *minimum confidence* dapat dikatakan sebagai valid *association rule* (David Samuel, 2008).

2.4 Algoritma Frequent Pattern Growth (FP-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 (David Samuel, 2008).

Pada penentuan *frequent itemset* terdapat 2 tahap proses yang dilakukan yaitu pembuatan *FP-tree* dan penerapan algoritma *FP-growth* untuk menemukan

frequent itemset. Struktur data yang digunakan untuk mencari *frequent itemset* dengan algoritma *FP-growth* adalah perluasan dari penggunaan sebuah pohon *prefix*, yang biasa disebut adalah *FP-tree*. Dengan menggunakan *FP-tree*, algoritma *FP-growth* dapat langsung mengekstrak *frequent itemset* dari *FP-tree* yang telah terbentuk dengan menggunakan prinsip *divide and conquer* (David Samuel, 2008).

2.4.1 Pencarian Association Menggunakan Algoritma FP-Growth

Proses pencarian *association rules* dengan menggunakan algoritma *FP-Growth* terdiri dari beberapa tahap yaitu:

1. Pembentukan *Fp-Tree*

FP-Tree merupakan struktur penyimpanan data yang dimampatkan. *FP-Tree* dibangun dengan memetakan setiap data transaksi kedalam setiap lintasan tertentu. Dalam *FP-Tree* mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data *FP-Tree* semakin efektif (David Samuel, 2008).

Pembangunan *FP-Tree* diawali dengan pembangunan tree pada setiap itemset. Pembangunan tree diawali dengan *prefix* berbeda, maka *itemset* berikutnya dibangun pada lintasan berbeda.

2. Penerapan Algoritma *FP-Growth*

Setelah tahap pembangunan *FP-Tree* maka selanjutnya akan diterapkan algoritma *FP-Growth* untuk mencari *Frequent Itemset* yang signifikan. Algoritma *FP-Growth* dibagi menjadi 3 langkah (David Samuel, 2008), yaitu:

a. Pembangkitan *Conditional Pattern Base*

Conditional Pattern Base merupakan sub *database* yang berisi *prefix path* (lintasan Prefix) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui *FP-Tree* yang telah dibangun sebelumnya.

b. Pembangkitan *Conditional FP-Tree*

Pada tahap ini, *support count* dari setiap item pada *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 Item*

Conditional FP-Tree merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi item untuk setiap *conditional FP-Tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-Growth* secara rekursif.

3. Pencarian *Association Rules*

Menurut (Jiawei dkk, 2012) proses pencarian *association rule* terbagi menjadi dua tahap yaitu analisis *frequent itemset* dan pembentukan *association rules*:

a. Analisis *Frequent Item*

Tahapan ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam database. Nilai *support* sebuah item diperoleh dengan persamaan rumus (2.4) dan untuk mencari nilai *support* dari dua item dapat menggunakan rumus (2.5).

b. Pembentukan *Assosiasi rule*

Setelah semua *frequent Itemset* ditemukan, kemudian dicari aturan *assosiasi* yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan *assosiatif* dengan menggunakan rumus (2.4). *Rules* yang telah didapat dilakukan pencocokan untuk memberikan saran kepada pengguna, dengan kata lain *rules* yang didapat akan diterjemahkan kedalam informasi yang dapat dimengerti oleh pengguna, yaitu berupa informasi.

2.5 *Assosiation Menggunakan Algoritma ECLAT*

Algoritma *Equivalence Class Transformation* (ECLAT) merupakan algoritma yang sangat sederhana untuk menemukan *itemset* yang paling sering muncul, pada dasarnya algoritma ECLAT melakukan pencarian secara *depth-first search* pada *database* dengan tata letak vertikal, jika *database* berbentuk horizontal maka harus dikonversikan ke bentuk vertikal terlebih dahulu.

Proses pencarian dilakukan dari item yang paling sering muncul hingga yang paling jarang muncul tanpa harus memperhatikan urutan, sehingga proses pemindaian tidak perlu dilakukan secara berulang-ulang. Data setiap *itemset* disimpan disebuah *Transaction ID List* (TID List), kemudian TID List diurutkan berdasarkan transaksi yang mengandung *itemset* yang sama (*frequent itemset*). Selanjutnya k-*itemset* diatur kedalam kelas-kelas berdasarkan kriteria tertentu yang

terbentuk dengan mempartisi suatu himpunan (*equivalence class*), $(k+1)$ -itemset bisa didapat dengan mengabungkan pasangan *frequent k-itemset* dari kelas yang sama. Dalam prosesnya, algoritma ini dilakukan secara rekursif, dimana pencarian itemset akan terus dilakukan sepanjang masih ada itemset yang tersisa (pencarian menyeluruh). Secara garis besar algoritma ECLAT dibagi menjadi 3 fase (Adita et al., n.d, 2014) yaitu:

1. Fase Inisialisasi: kontruksi perhitungan global untuk *frequent 2-itemset*.
2. Fase Transformasi ; mempartisi *frequent 2-itemset* dan menjadwalkan partisi atas *processor*. Fase ini juga melakukan vertikal tranformasi pada *database*.
3. Fase *Asynchronus*: kontruksi frequent k-itemset, sesuai tujuan yang diinginkan.

2.5.1 Pencarian Association Menggunakan Algoritma ECLAT

Menurut (Adita et al., n.d, 2014), proses pembentukan itemset pada algoritma ECLAT ada beberapa tahapan yaitu:

1. Dimulai dengan merepresentasikan data ke dalam bentuk format data vertikal, jika item pada transaksi berbentuk horizontal maka diubah terlebih dahulu menjadi bentuk vertikal dengan menggabungkan TID List pada transaksi yang memiliki item yang sama.
2. Tahapan selanjutnya dilakukan penyilangan pada TID List, contoh: untuk mencari penyilangan 2 itemset dilakukan penyilangan subset item X dan subset item Y, proses penyilangan terus dilakukan secara rekursif sampai tidak ada lagi itemset yang tersisa.
3. Tahapan selanjutnya tentukan minimum support (minsup) dari setiap k-itemset. Itemset yang memiliki nilai kurang dari nilai minsup yang ditentukan maka akan dihilangkan dari proses penyilangan tersebut. sehingga didapatkan hasil frequent dari 2 itemset.
4. Setelah semua pola *frequent Itemset* ditemukan, kemudian dicari aturan assosiasi yang memenuhi syarat minimum dengan menghitung confidence aturan assosiatif dengan menggunakan rumus (2.6). Rules yang telah didapat dilakukan pencocokan untuk memberikan saran kepada pengguna, dengan kata lain rules yang didapat akan diterjemahkan kedalam informasi yang dapat dimengerti oleh pengguna, yaitu berupa informasi.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

State Islamic University of Sultan Syarif Kasim Riau

2.6 Kajian Pustaka

Berikut adalah beberapa penelitian sebelumnya yang menjadi acuan dan dasar penelitian ini :

Tabel 2.1 Kajian Pustaka

No.	Peneliti (tahun)	Metode yang digunakan	keterangan
1	(Fatihatul dkk, 2013)	FP-Growth	hasil yang didapat adalah perusahaan dapat menggunakan hasil rules tersebut dalam membuat strategi-strategi untuk meningkatkan penjualan dengan menghasilkan nilai strong confidence paling tinggi 92 %.
2	(Mehay dkk, 2013)	FP-Growth Apriori	Selama proses penelitian dengan menggunakan jumlah data transaksi yang sama ditemukan bahwasanya algoritma FP-Growth lebih cepat pada data transaksi yang cukup besar dibandingkan dengan apriori dan selain itu Fp-Growth juga memerlukan waktu komputasi yang sedikit untuk mendapatkan frequent itemset.
3	(Kaur dan Singh, 2013)	FP-Growth	FP-Growth lebih efisien dibandingkan dengan algoritma assosiasi yang lain dalam mencari frequent pattern pada sebuah database yang besar dikarenakan FP-Growth bekerja dalam mencari dataset dan menghasilkan sebuah assosiasi rule yang bertujuan untuk mendapatkan hasil yang maksimal.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4	(Singh dkk, 2014)	Apriori FP-Growth	Apriori memerlukan waktu yang lama dalam komputasi asosiasi rule pada sebuah data transaksi yang sama banyak dan FP-Growth jauh lebih cepat dibandingkan dengan Apriori dikarenakan tidak memerlukan generate candidate untuk mendapatkan frequent itemsets, dan Hanya membutuhkan dua kali scanning database dalam mencari frequent itemsets.
5	(Trupti dkk, 2014)	AIS, SETM, Apriori, Aprioritid, Apriori Hybrid, FP-Growth	Berdasarkan dari beberapa parameter seperti waktu eksekusi, data support, dan eksekusi didapatkan kesimpulan bahwasanya FP-growth lebih baik dari beberapa algoritma yang diuji pada penelitian ini.

Hak cipta milik UIN Suska Riau

State Islamic University of Sultan Syarif Kasim Riau

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.