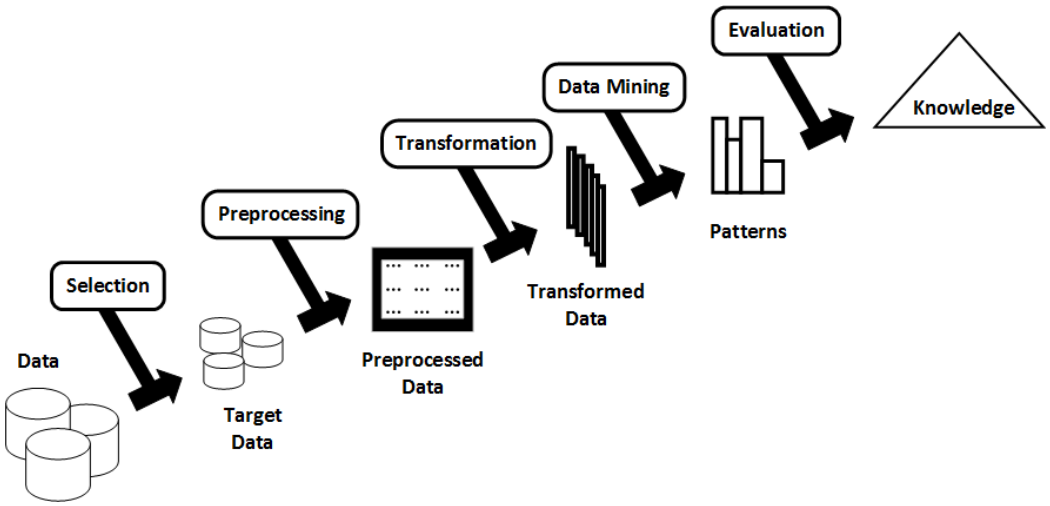
BAB II

KAJIAN PUSTAKA

2.1. *Knowladge Discovery in Database* (KDD)

Knowledge Discovery in Database (KDD) adalah proses menentukan informasi yang berguna serta pola-pola yang ada dalam data. Informasi ini terkandung dalam basis data yang berukuran besar yang sebelumnya tidak diketahui dan potensial bermanfaat. *Data mining* merupakan salah satu langkah dari serangkaian proses *iterative* KDD (Ikhwan et al., 2015). Proses KDD secara garis besar dapat dijelaskan sebagai berikut (Kusrini & Luthfi, 2009:7):



Gambar 2.1 Proses KDD
Sumber: Hermawati (2013:6)

1. *Data Selection*

   Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing / Cleaning*

   Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*). Juga dilakukan proses *enrichment*, yaitu proses memperkaya data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. *Transformation*

   Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data Mining*

   Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau

algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation / Evaluation*

   Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.2. *Data Mining*

2.2.1. Sejarah *Data Mining*

Nama *data mining* sebenarnya mulai dikenal sejak tahun 1990, ketika pekerjaan pemanfaatan data menjadi sesuatu yang penting dalam berbagai bidang, mulai dari bidang akademik, bisnis, hingga medis. *Data mining* dapat diterapkan pada berbagai bidang yang mempunyai sejumlah data, tetapi karena wilayah penelitian dengan sejarah yang belum lama, dan belum melewati masa remaja, maka *data mining* diperdebatkan posisi bidang pengetahuan yang memilikinya. Maka Daryl Pregibon menyatakan bahwa *data mining* adalah campuran dari statistik, kecerdasan buatan, dan riset basis data yang masih berkembang (Gorunescu, 2011 dalam buku Prasetyo, 2014:1).

Terlepas dari remajanya *data mining*, ternyata *data mining* diproyeksikan menjadi jutaan dolar di dunia industri pada tahun 2000, sedangkan pada saat yang sama, ternyata *data mining* dipandang sebelah mata oleh sejumlah peneliti sebagai

dirty word in statistics. Mereka adalah orang-orang yang tidak memandang *data mining* sebagai sesuatu yang menarik bagi mereka pada saat itu (Gorunescu, 2011 dalam buku Prasetyo, 2014:1).

Munculnya *data mining* didasarkan pada jumlah data yang tersimpan dalam basis data semakin besar. Misalnya dalam sebuah supermarket, ada berapa transaksi pelanggan yang terjadi dalam sehari dan ada berapa juta data yang sudah tersimpan dalam sebulan. Dalam perusahaan, ada berapa juta data yang sudah tersimpan dari setiap kegiatan produksi untuk setiap produk yang dibuat dalam beberapa tahun. Contoh lain, jika Anda mempunyai kartu kredit, mungkin Anda sering menerima surat penawaran barang dan jasa. Jika bank mempunyai 1.000.000 nasabah dan biaya pengiriman surat per nasabah adalah 500 rupiah, maka biaya yang harus dikeluarkan bank adalah 500 juta rupiah., padahal nasabah yang mungkin benar-benar membeli hanya sekitar 15%. Akibatnya ada pembuangan biaya sekitar 85% dari 500 juta atau sekitar 425 juta; sungguh sia-sia. Jika perusahaan dapat memanfaatkan data-data yang ada sehingga hanya nasabah yang berpotensi untuk membeli saja yang dikirim surat, maka biaya pengiriman tersebut dapat ditekan (Prasetyo, 2014:1-2).

Kemajuan luar biasa yang terus berlanjut dalam bidang *data mining* didorong oleh beberapa faktor, antara lain (Larose, 2005 dalam buku Kusrini & Luthfi, 2009:4):

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam data *warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang andal.

3. Adanya peningkatan akses data melalui navigasi *web* dan intranet.
4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk *data mining* (ketersediaan teknologi).

2.2.2. Pengertian *Data Mining*

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *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 terakit dari berbagai *database* besar (Turban, dkk. 2005 dalam buku Kusrini & Luthfi, 2009:3).

Data mining merupakan analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data dengan cara yang berbeda dengan sebelumnya, yang dapat dipahami dan bermanfaat bagi pemilik data. *Data mining* merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, *database*, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari *database* yang besar (Larose, 2005 dalam buku Kusrini & Luthfi, 2009:4).

Data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. *Data mining* adalah analisis otomatis dari data yang berjumlah besar atau

kompleks dengan tujuan untuk menemukan pola atau kecenderungan yang penting yang biasanya tidak disadari keberadaannya (Pramudiono, 2006 dalam buku Kusrini & Luthfi, 2009:4).

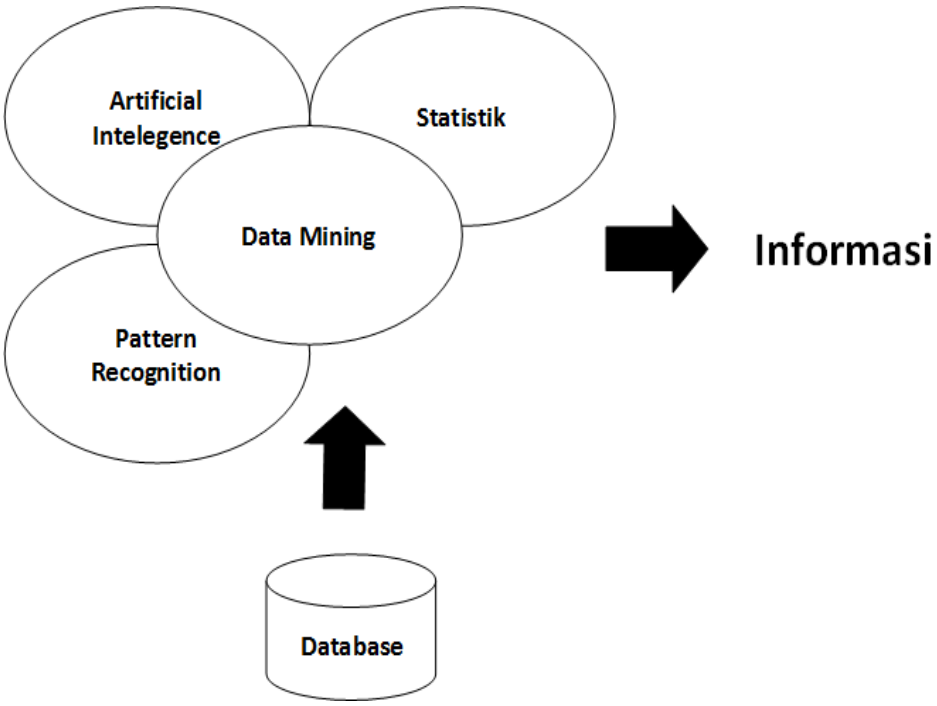
Data mining adalah proses mempekerjakan satu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis. Definisi lain diantaranya adalah pembelajaran berbasis induksi (*induction-based learning*) adalah proses pembentukan definisi-definisi konsep umum yang dilakukan dengan cara mengobservasi contoh-contoh spesifik dari konsep-konsep yang akan dipelajari. *Knowledge Discovery In Databases* (KDD) adalah penerapan metode saintifik pada *data mining*. Dalam konteks ini *data mining* merupakan suatu langkah dari proses KDD (Hermawati, 2013).

Data mining adalah proses untuk mendapatkan informasi dengan melakukan pencarian pola dan relasi-relasi yang tersembunyi di dalam timbunan data yang banyak (Ikhwan et al., 2015).

Berdasarkan definsi-definisi yang disampaikan di atas, dapat disimpulkan bahwa *data mining* adalah suatu proses otomatis yang digunakan untuk menemukan hubungan atau pola dalam jumlah data yang sangat besar yang dapat dipahami dan bermanfaat bagi pemilik data yang biasanya tidak menyadari akan keberadaan informasi tersebut.

2.2.3. Akar Ilmu *Data Mining*

Ada istilah lain yang mempunyai makna yang sama dengan *data mining* yaitu *knowledge discovery in database* (KDD). Memang *data mining* atau KDD bertujuan untuk memanfaatkan data dalam basis data dengan mengolahnya sehingga menghasilkan informasi baru yang berguna. Seperti dilustrasikan pada Gambar 2.2, jika dilacak akar keilmuanya, ternyata *data mining* mempunyai empat akar bidang ilmu sebagai berikut (Prasetyo, 2014:2-3):



Gambar 2.2 Akar Ilmu *Data Mining*
Sumber: Prasetyo (2014:2)

1. Statistik

   Bidang ini merupakan akar paling tua, tanpa ada statistik maka *data mining* mungkin tidak ada. Dengan menggunakan statistik klasik ternyata data yang diolah dapat diringkas dalam apa aja yang umum dikenal sebagai *exploratory*

data analysis (EDA). EDA berguna untuk mengidentifikasikan hubungan sistemastis antar variabel/fitur ketika tidak ada cukup informasi alami yang dibawanya

2. Kecerdasan Buatan atau *Artifical Intelligence* (AI)

   Bidang ilmu ini berbeda dengan statistik. Teorinya dibangun berdasarkan teknik heuristik sehingga AI berkontribusi terhadap teknik pengolahan informasi berdasarkan pada model penalaran manusia. Salah satu cabang dari AI, yaitu pembelajaran mesin atau *machine learning*, merupakan disiplin ilmu yang paling penting yang direpresentasikan dalam pembangun *data mining*, menggunakan teknik di mana sistem komputer belajar dengan pelatihan.

3. Pengenalan Pola

   Sebenarnya *data mining* juga menjadi turunan bidang pengenalan pola, tetapi hanya mengolah data dari basis data. Data yang diambil dari basis data untuk diolah bukan dalam bentuk relasi, melainkan dalam bentuk normal pertama sehingga set data dibentuk menjadi bentuk normal pertama. Akan tetapi, *data mining* mempunyai ciri khas yaitu pencarian pola asosiasi dan pola sekuensial.

4. Sistem Basis Data

   Akar bidang ilmu keempat dari *data mining* yang menyediakan informasi berupa data yang akan digali menggunakan metode-metode yang disebutkan sebelumnya.

2.2.4. Proses *Data Mining*

Secara sistematis, ada tiga langkah utama dalam *data mining* (Gorunescu, 2011 dalam buku Prasetyo, 2014:7):

1. Eksplorasi / Pemrosesan Awal Data

   Eksplorasi / pemrosesan awal data terdiri dari pembersihan data, normalisasi data, transformasi data, penanganan data yang salah, reduksi dimensi, pemilihan subset fitur, dan sebagainya.

2. Membangun Model dan Melakukan Validasi Terhadapnya

   Membangun model dan melakukan valisasi terhadapnya berarti melakukan analisis berbagai model dan memilih model dengan kinerja prediksi yang terbaik. Dalam langkah ini digunakan metode-metode seperti klasifikasi, regresi, analisis *cluster*, deteksi *anomaly*, analisis asosiasi, analisi pola sekuensial, dan sebagainya. Dalam beberapa referensi deteksi anomali juga masuk dalam langkah eksplorasi. Akan tetapi, deteksi anomali juga dapat digunakan sebagai algoritma utama, terutama untuk mencari data-data yang spesial.

3. Penerapan

   Penerapan berarti menerapkan model pada data yang baru untuk menghasilkan perkiraan / prediksi masalah yang diinvestigasi.

2.2.5. CRISP-DM

Chross-Industry Standard Process for Data Mining (CRISP-DM) yang dikembangkan tahun 1996 oleh analis dari beberapa industry seperti Daimler Chrysler, SPSS, dan NCR. CRISP-DM menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Dalam CRISP-DM, sebuah proyek *data mining* memiliki siklus hidup yang terbagi dalam enam fase (Gambar 2.3). Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan penting antar fase digambarkan dengan panah. Sebagai contoh, jika proses berada pada fase *modelling*. Berdasarkan pada perilaku dan karakteristik model, proses mungkin harus kembali kepada fase *data preparation* untuk perbaikan lebih lanjut terhadap data atau berpindah maju kepada fase *evaluation*. (Kusrini & Luthfi, 2009:8)

Gambar 2.3 Proses *Data Mining* Menurut CRISP-DM
Sumber: Kusrini & Luthfi (2009:8)

Enam fase CRISP-DM (Larose, 2005 dalam buku Kusrini & Luthfi, 2009:9-10), yaitu:

1. Fase Pemahaman Bisnis *(Business Understanding Phase)*
    a. Penentuan tujuan proyek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.
    b. Menerjemahkan tujuan dan batasan menjadi formula dari permasalahan *data mining*.
    c. Menyiapkan strategi awal untuk mencapai tujuan.
2. Fase Pemahaman Data *(Data Understanding Phase)*
    a. Mengumpulkan data.
    b. Menggunakan analisis penyelidikan data untuk mengenali lebih lanjut data dan pencarian pengetahuan awal.
    c. Mengevaluasi kualitas data.
    d. Jika diinginkan, pilih sebagian kecil group data yang mungkin mengandung pola dari permasalahan.
3. Fase Pengolahan Data *(Data Preparation Phase)*
    a. Siapkan dari data awal, kumpulan data yang akan digunakan untuk keseluruhan fase berikutnya. Fase ini merupakan pekerjaan berat yang perlu dilaksanakan secara insentif.
    b. Pilih kasus dan variabel yang ingin dianalisis dan yang sesuai analisis yang akan dilakukan.
    c. Siapkan data awal sehingga siap untuk perangkat pemodelan.

4. Fase Pemodelan *(Modeling Phase)*
    a. Pilih dan aplikasikan teknik pemodelan yang sesuai.
    b. Kalibrasi aturan model untuk mengoptimalkan hasil.
    c. Perlu diperhatikan bahwa beberapa Teknik mungkin untuk digunakan pada permasalahan *data mining* yang sama.
    d. Jika diperlukan, proses dapat kembali ke fase pengolahan data untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik *data mining* tertentu.
5. Fase Evaluasi *(Evaluation Phase)*
    a. Mengevaluasi satu atau lebih model yang digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas sebelum disebarkan untuk digunakan.
    b. Menetapkan apakah terdapat model yang memenuhi tujuan pada fase awal.
    c. Menentukan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik.
    d. Mengambil keputusan berkaitan dengan penggunaan hasil dari *data mining*.
6. Fase Penyebaran *(Deployment Phase)*
    a. Menggunakan model yang dihasilkan. Terbentuknya model tidak menandakan telah terselesaikannya proyek.
    b. Contoh sederhana penyebaran: Pembuatan laporan

c. Contoh kompleks penyebaran: Penerapan proses *data mining* secara paralel pada departemen lain.

2.2.6. Pengelompokan *Data Mining*

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu (Larose, 2015 dalam buku Kusrini & Luthfi, 2009:10-12):

1. Deskripsi

   Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indekx berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi

dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

3. Prediksi

   Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Contoh prediksi dalam bisnis dari penelitian adalah prediksi harga beras dalam tiga bulan yang akan datang atau prediksi presentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikan. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

4. Klasifikasi

   Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah. Contoh lain klasifikasi dalam bisnis dan penelitian adalah menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan, memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk, atau mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran

   Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang

lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lainnya. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel taget. Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Contoh pengklusteran dalam bisnis dan penelitian adalah mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.

6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja *(market basket analysis)*. Contoh asosiasi dalam bisnis dan penelitian adalah meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan atau menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

2.2.7. Tantangan Dalam *Data Mining*

Tantangan dalam *data mining* meliputi (Hermawati, 2013:19):

1. *Scalability*, yaitu besarnya ukuran basis data yang digunakan.
2. *Dimensionality*, yaitu banyaknya jumlah atribut dalam data yang akan diproses.
3. *Complex and Heterogeneous Data*, yaitu data yang kompleks dan mempunyai variasi yang beragam.
4. *Data Quality*, yaitu kualitas data yang akan diproses seperti data yang bersih dari *noise*, *missing value*, dsb.
5. *Data Ownership and Distribution*, yaitu siapa yang memiliki data dan bagaimana distribusinya.
6. *Privacy Preservation*, yaitu menjaga kerahasiaan data yang banyak diterapkan pada data nasabah perbankan.
7. *Streaming Data*, yaitu aliran data itu sendiri.

2.3. Metode *Data Mining*

2.3.1. Algoritma *FP-Growth* (*Frequent Pattern Growth*)

FP-Growth adalah salah satu *alternative* algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. *FP-Growth* menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada algoritma Apriori (Gunadi & Sensuse, 2012).

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* (Gunadi & Sensuse, 2012).

2.3.1.1. Pembuatan *FP-Tree*

FP-Tree merupakan struktur penyimpanan data yang dimampatkan. *FP-Tree* dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam *FP-tree*. Karena dalam setiap transaksi yang dipetakan, 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 (Gunadi & Sensuse, 2012).

Adapun *FP-Tree* adalah sebuah pohon dengan definisi sebagai berikut (Gunadi & Sensuse, 2012):

1. *FP-Tree* dibentuk oleh sebuah akar yang diberi label *null*, sekumpulan *sub-tree* yang beranggotakan item-item tertentu, dan sebuah tabel *frequent header*.
2. Setiap simpul dalam *FP-Tree* mengandung tiga informasi penting, yaitu label item, menginformasikan jenis *item* yang direpresentasikan simpul tersebut, *support count*, merepresentasikan jumlah lintasan transaksi yang melalui simpul tesebut, dan pointer penghubung yang menghubungkan simpul-

simpul dengan label *item* sama antar-lintasan, ditandai dengan garis panah putus-putus.

2.3.1.2. Penerapan Algoritma *FP-Growth*

Setelah tahap pembangunan *FP-Tree* dari sekumpulan data transaksi, akan diterapkan algoritma *FP-Growth* untuk mencari *frequent itemset* yang signifikan. Algoritma *FP-Growth* dibagi menjadi tiga langkah utama, yaitu (Gunadi & Sensuse, 2012):

1. Tahap 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.

2. 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 ξ akan dibangkitkan dengan *conditional FP-Tree*.

3. Tahap Pencarian *Frequent Itemset*

   Apabila *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

2.3.2. *Association Rule*

Analisis asosiasi atau *association rule mining* adalah teknik *data mining* untuk menemukan aturan asosiasi antara suatu kombinasi item. *Interestingness measure* yang dapat digunakan dalam data mining adalah (Gunadi & Sensuse, 2012):

A. *Support*, adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu item atau itemset dari keseluruhan transaksi.

B. *Confidence*, adalah suatu ukuran yang menunjukkan hubungan antar dua item secara *conditional* (berdasarkan suatu kondisi tertentu).

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap (Ikhwan et al., 2015):

1. Analisa Pola Frekuensi Tinggi

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

$$Support\ (A) = \frac{\text{Jumlah Transaksi Mengandung A}}{\text{Total Transaksi}}$$

Rumus 2.1 *Support*

$$Support\ (A \cap B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi}}$$

Rumus 2.2 *Support A C B*

2. Pembentukan Aturan Assosiatif

   Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan assosiatif yang memenuhi syarat minimum untuk *confidence* dengan

menghitung *confidence* aturan assosiatif A → B. Nilai *confidence* dari aturan A → B diperoleh dari rumus berikut:

$$\textit{Confidence}\ (A \rightarrow B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi Mengandung A}}$$

Rumus 2.3 *Confidence (A→B)*

Lift ration adalah parameter penting selain *support* dan *confidence* dalam *association rule*. *Lift ration* mengukur seberapa penting rule yang telah terbentuk berdasarkan nilai *support* dan *confidence*. Lift Ratio merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B. *Improvement Ratio* dapat dihitung dengan rumus (Ikhwan et al., 2015):

$$\textit{Lift Ration} = \frac{\text{Support } (A \cap B)}{\text{Support A} * \text{Support B}}$$

Rumus 2.4 *Lift Ration*

Sebuah transaksi dikatakan valid jika mempunyai nilai *lift ration* lebih dari atau sama dengan 1, yang berarti bahwa dalam transaksi tersebut item A dan item B benar-benar dibeli secara bersamaan.

2.4. *Software* Pendukung

2.4.1. WEKA 3.6

Menurut situs resmi WEKA (2018), Weka adalah kumpulan algoritma pembelajaran mesin untuk tugas-tugas penambangan data. Algoritma dapat diterapkan langsung ke kumpulan data atau dipanggil dari kode Java Anda sendiri. Weka berisi alat untuk data *pre-processing, classification, regression, clustering, association rules, and visualization*. Ini juga sangat cocok untuk mengembangkan

skema pembelajaran mesin baru. Ditemukan hanya di pulau-pulau Selandia Baru, Weka adalah burung yang tak bisa terbang dengan sifat ingin tahu. Namanya diucapkan seperti ini, dan suara burung seperti ini. Weka adalah perangkat lunak *open source* yang diterbitkan di bawah *General Public License* GNU. WEKA 3.6 merupakan salah satu versi dari WEKA yang akan digunakan dalam penelitian ini.

2.4.2. PHP 7

PHP (PHP *Hypertext Prepocessor*) adalah salah bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutkan akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server*, PHP dikatakan sebagai bahasa sisi *server* (*server-side*). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat *user* memilih perintah *View Source* pada *web browser* yang mereka gunakan (Raharjo, 2016:38).

Beberapa kelebihan PHP antara lain (Hidayatullah & Kawistara, 2015:234-237):

1. PHP berbasis *server-side scripting*.
2. *Command Line Scripting* pada PHP.
3. PHP dapat membuat aplikasi desktop.
4. Digunakan untuk berbagai macam *platform* OS.
5. Mendukung berbagai macam *web server*.

6. *Object Oriented Programming* atau *Procedural*.
7. *Output file* PHP pada XHTML, HTML, & XML.
8. Mendukung banyak RDMS (*database*).
9. Mendukung banyak komunikasi.
10. Pengelolahan teks yang sangat baik.

PHP 7 baru diliris pada bulan Desember 2015, sebagai penerus dari versi sebelumnya, yaitu PHP 5.6. Rilis baru tersebut dinamai dengan PHP 7, bukan 5.7. PHP 7 menjanjikan performansi yang lebih cepat disbanding versi-versi sebelumnya. Dalam PHP 7, ditambahkan beberapa fitur-fitur baru seperti pembuatan *array* konstan menggunakan fungsi define(), versi baru penggunaan kata kunci *use* pada saat menyertakan suatu *namespace*, pendeklarasian tipe parameter fungsi, pendefinisian tipe kembalian suatu fungsi, dan lain-lain (Raharjo, 2016:387).

2.4.3. *MySQL*

MySQL (My Structured Query Language) adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* AB membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL (Solichin. Achmad, 2016:85).

Tidak seperti *Apache* yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu *MySQL AB*. *MySQL* AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL* AB adalah: David Axmark, Allan Larsson, dan Michael Monty Widenius. Beberapa kelebihan *MySQL* antara lain (Solichin. Achmad, 2016:85):

1. *Free* (bebas di*download*)
2. Stabil dan tangguh
3. Fleksibel dengan berbagai pemrograman
4. Security yang baik
5. Dukungan dari banyak komunitas
6. Kemudahan management *database*.
7. Mendukung transaksi
8. Perkembangan software yang cukup cepat.

2.4.4. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (localhost), yang terdiri atas program *Apache* HTTP *Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, PHP dan *Perl*. Program ini tersedia dalam

GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis (Aditya, 2011:16) (Idayani, Sutardi, & Muchlis, 2017).

2.4.5. UML

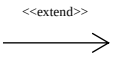
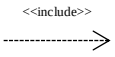
Pada perkembangan teknik pemrogramaan berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrogramaan berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa, A.S Shalahuddin, 2015:137).

2.4.5.1. *Use Case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan 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. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* (Rosa, A.S Shalahuddin, 2015:155).

Berikut adalah simbol-simbol yang ada pada diagram *use case* (Rosa, A.S Shalahuddin, 2015:156-158):

Tabel 2.1 Simbol-Simbol Yang Ada Pada *Use Case Diagram*

Simbol	Nama	Deskripsi
	*Use Case*	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
	Aktor / *Actor*	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun symbol 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 berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor
	Extensi / *Extend*	Relasi *use case* tambahan ke sebuah *use case* di mana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu
	Generalisasi / *Generalization*	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
	*Include*	Relasi *use case* tambahan ke sebuah *use case* di mana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini

Sumber: Rosa, A.S Shalahuddin (2015:156-158)

2.4.5.2. *Activity Diagram*

Diagram aktivitas atau *activiy* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa, A.S Shalahuddin, 2015:161).

Berikut adalah simbol-simbol yang ada pada diagram *activity* (Rosa, A.S Shalahuddin, 2015:162-163):

Tabel 2.2 Simbol-Simbol Yang Ada Pada *Activity Diagram*

Simbol	Nama	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
	Percabangan / *Decision*	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Penggabungan / *Join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	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

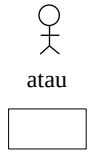
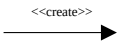
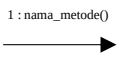
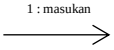
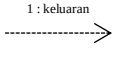
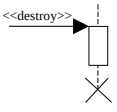
Sumber: Rosa, A.S Shalahuddin (2015:162-163)

2.4.5.3. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefiniskan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (Rosa, A.S Shalahuddin, 2015:165).

Berikut adalah simbol-simbol yang ada pada diagram *class* (Rosa, A.S Shalahuddin, 2015:165-167):

Tabel 2.3 Simbol-Simbol Yang Ada Pada *Sequence Diagram*

Simbol	Nama	Deskripsi
	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal *frase* nama aktor
	Garis Hidup / *Lifetime*	Menyatakan kehidupan suatu objek
	Objek	Menyatakan objek yang berinteraksi pesan
	Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
	Pesan Tipe *Create*	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
	Pesan Tipe *Call*	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
	Pesan Tipe *Send*	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi
	Pesan Tipe *Return*	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
	Pesan Tipe *Destroy*	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*

Sumber: Rosa, A.S Shalahuddin (2015:165-167)

2.4.5.4. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa, A.S Shalahuddin, 2015:141-142).

Berikut adalah simbol-simbol yang ada pada diagram *class* (Rosa, A.S Shalahuddin, 2015:146-147):

Tabel 2.4 Simbol-Simbol Yang Ada Pada *Class Diagram*

Simbol	Nama	Deskripsi
	Kelas	Kelas pada struktur sistem
	Antramuka / *Interface*	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
	Asosiasi / *Association*	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
	Asosiasi Berarah / *Directed Association*	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
	Generalisasi / *Generalization*	Relasi antarkelas dengan makna generalisasi-spesialisasi(umum-khusus)
	Kebergantungan / *Dependency*	Relasi antarkelas dengan makna kebergantungan antarkelas
	Agregasi / *Aggregation*	Relasi antarkelas dengan makna semua-bagian *(whole-part)*

Sumber: Rosa, A.S Shalahuddin (2015:146-147)

2.5. Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam

mengkaji penelitian yang dilakukan. Berikut beberapa penelitian yang menjadi referensi dalam penelitian tersebut:

1. **Meilani & Azmuri (2015)** dengan jurnal ISSN 2089-1121 yang berjudul "Penentuan Pola Yang Sering Muncul Untuk Penerima Kartu Jaminan Kesehatan Masyarakat (JAMKESMAS) Menggunakan Metode *FP-Growth*", yang menghasilkan kesimpulan bahwa: Dengan menggunakan algoritma *fp-growth* dapat menghasilkan pola-pola yang sering muncul pada penerima kartu jamkesmas berdasarkan kriteria miskin antara lain luas lantai, jenis lantai, jenis dinding, fasilitas bab, sumber air, bahan bakar masak, pendapatan, pendidikan, aset; Berdasarkan dari hasil pola yang didapat akan digunakan untuk acuan atau informasi sebagai data penunjang pengambilan keputusan penerima kartu jamkesmas; Minimum *support* diatas 50% tidak menghasilkan pola penerima kartu jamkesmas; Dari hasil analisis pola yang sering muncul menunjukkan bahwa penerima kartu jamkesmas pada kelurahan tersebut yang tepat sasaran sekitar 60% dikarenakan masih ada penduduk yang kurang memenuhi kriteria miskin.
2. **Fitriyani (2015)** dengan jurnal ISSN 2355-6579 yang berjudul "Implementasi Algoritma *FP-Growth* Menggunakan *Association Rule* Pada *Market Basket Analysis*", yang menghasilkan kesimpulan bahwa: Algoritma *FP-Growth* lebih efisien dalam memproses market basket atau data keranjang belanja konsumen; Menggunakan *association rule* dapat mengetahui *rule-rule* belanja konsumen seperti hubungan-hubungan tiap produk yang dibeli oleh konsumen dan mempengaruhi tipe-tipe konsumen dengan produk yang

dibeli; *Association Rule* dapat memberikan keputusan pada manajemen untuk *display* produk, promosi, strategi pemasaran dan lain-lain.

3. **Larasati, Nasrun, & Ahmad (2015)** dengan jurnal ISSN 2355-9365 yang berjudul "Analisis dan Implementasi Algoritma *FP-Growth* Pada Aplikasi Smart Untuk Menentukan *Market Basket Analysis* Pada Usaha Retail (Studi Kasus: PT. X)", yang menghasilkan kesimpulan bahwa: Pengimplementasian perancangan *Market Basket Analysis* dengan menggunakan *FP-Growth* di aplikasi SMART sudah berhasil; Pengimplementasian algoritma *FP-Growth* di aplikasi SMART sudah tervalidasi untuk melakukan *Market Basket Analysis*; *Rules* yang dihasilkan di setiap bulan dengan *minsupp*=0.002 dan *minconf*=0.5, jumlahnya hampir sama dan ada beberapa produk yang muncul di beberapa bulan yang berbeda; Semakin besar *minsupp* dan *minconf* yang dimasukkan, maka menghasilkan *rule* yang semakin sedikit. Rata-rata dengan nilai *minsupp*=0.006 dan *minconf* = 0.6 tidak menghasilkan *rules*; Jumlah *dataset* tidak mempengaruhi waktu proses dalam analisis ini. Rata-rata waktu proses pada analisis ini adalah 957 ms.
4. **Idayani et al. (2017)** dengan jurnal ISSN 2502-8928 yang berjudul "Perancangan Aplikasi Data Warehouse Menggunakan Metode *FP-Growth* Untuk Memprediksi Penjualan Alat-Alat Kesehatan (Studi Kasus : Apotek Kimia Farma Korem)", yang menghasilkan kesimpulan bahwa: Metode dalam pencarian *Frequent itemsset* menggunakan algoritma *FP-Growth* bekerja sangat baik dalam melakukan *Frequent itemsset* dengan proses pembentukan *FP-Tree* dengan menghasilkan *rule* dari data penjualan alat-alat

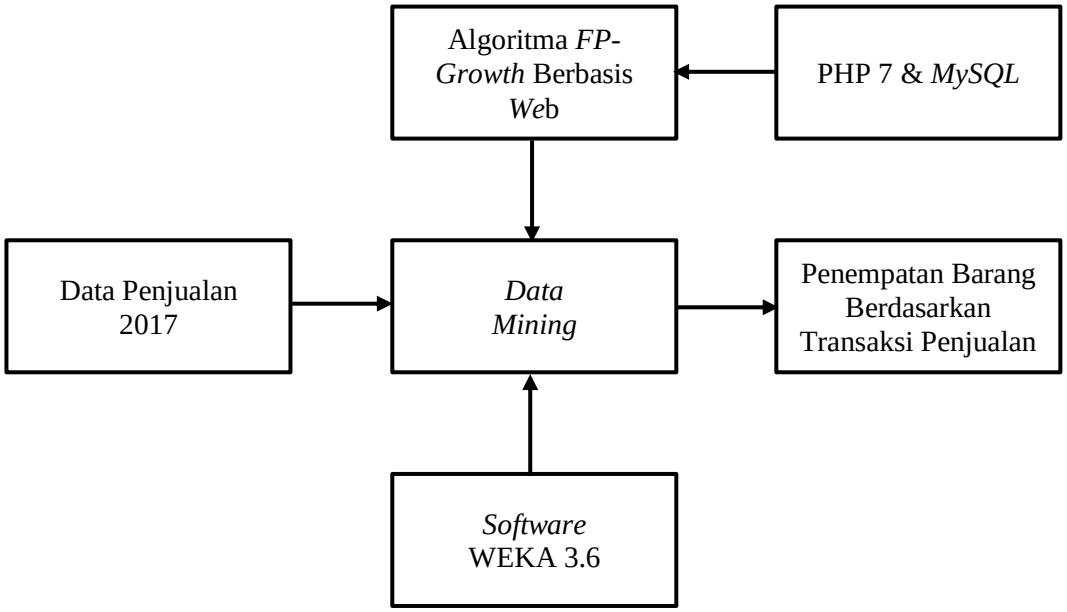
kesehatan; Dalam mengimplementasikan algoritma *FP-Growth* pada aplikasi prediksi persediaan alat-alat kesehatan dapat dilihat dari banyaknya item yang terjual, lalu membuat rangkaian *tree* dengan *FP-Tree* untuk mengetahui banyaknya frequent itemset yang terjadi; Algoritma *FP-Growth* dapat diterapkan untuk mendukung strategi penjualan alat-alat kesehatan pada Apotik Kimia Farma Korem sehingga pihak manajemen dari Apotik itu sendiri dapat melakukan pengambilan keputusan dengan cepat.

5. **Vijayarani & R. (2015)** dengan jurnal internasional ISSN 2320-9801 yang berjudul "*Association Rule Generation in Data Streams using FP-Growth and Apriori MR Algorithms*". *The data stream is continuous arrival of data and this data is normally dynamic in nature and its size is very huge. In this situation, it is not possible to perform data mining tasks with the traditional algorithms since those algorithms are suitable for static data bases. Hence, data stream mining needs the development of new algorithms and techniques to perform the data mining tasks. Association rules are described by finding the frequent pattern, links, relationship and the related structures among the data objects in the databases. There are two important steps in association rule mining; first step is to find the frequent data items and second step is to generate association rules from the frequent data items. This work has compared two different types of association rule mining algorithms; they are APRIORI Map/Reduce algorithm and Frequent Pattern Tree Growth algorithm.* F*rom the experimental results, it is observed that the performance of FP-Tree Growth algorithm is efficient than APRIORI MR Algorithm*.

2.6. Kerangka Pemikiran

Kerangka berpikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah penting. Kerangka berpikir yang baik akan menjelaskan secara teoritis pertautan antar variabel yang akan di teliti (Sugiyono, 2014).

Adapun kerangka pemikiran dari penelitian ini sebagai berikut:



Gambar 2.4 Kerangka Pemikiran
Sumber: Data Olahan Peneliti (2018)