

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Siyamto (2017) mengenai Pemanfaatan *Data mining* Dengan Metode *Clustering* Untuk Evaluasi Biaya Dokumen Ekspor Di PT Winstar Batam, Siyamto menggunakan *data mining* menggunakan metode *clustering* untuk mengevaluasi biaya dokumen ekspor, sedangkan penelitian ini menggunakan metode *clustering* untuk mengetahui minat *customer* pada UD. Syahril.

Berdasarkan penelitian yang dilakukan oleh Syahputra, dkk (2018) mengenai Penerapan *Data mining* Dalam Menentukan Pilihan Jurusan Bidang Studi SMA Menggunakan Metode *Clustering* Dengan Teknik *Single Linkage*, Syahputra, dkk menggunakan metode *clustering* untuk menentukan pilihan jurusan, sedangkan penelitian ini menggunakan metode *clustering* untuk mengetahui minat *customer* pada UD. Syahril.

Berdasarkan penelitian yang dilakukan oleh Silvi (2018) mengenai Analisis *Cluster* dengan *Data Outlier* Menggunakan *Centroid Linkage* dan *K-Means Clustering* untuk Pengelompokan Indikator HIV/AIDS di Indonesia, Silvi menggunakan *clustering* untuk mengelompokkan indikator HIV/AIDS, sedangkan penelitian ini menggunakan metode *clustering* untuk mengetahui minat *customer* pada UD. Syahril.

Berdasarkan penelitian yang dilakukan oleh Astuti (2017) mengenai Penerapan *Data mining* Untuk *Clustering* Data Penduduk Miskin Menggunakan Algoritma *Hard C-Means*, Astuti menggunakan *clustering* untuk data penduduk miskin, sedangkan penelitian ini menggunakan metode *clustering* untuk mengetahui minat *customer* pada UD. Syahril.

Berdasarkan penelitian yang dilakukan oleh Metisen dan Sari (2015) mengenai Analisis *Clustering* Menggunakan Metode *K-Means* Dalam Pengelompokkan Penjualan Produk Pada Swalayan Fadhila, Metisen dan Sarimenggunakan metode *K-Means* untuk pengelompokan data penjualan produk sedangkan penelitian ini menggunakan metode *clustering* untuk mengetahui minat *customer* pada UD. Syahril.

II.2. Landasan Teori

Berikut ini adalah beberapa landasan teori yang digunakan dalam pembuatan skripsi ini.

II.2.1. *Data mining*

Data mining adalah teknik analisa data secara otomatis untuk membuka atau membongkar hubungan dari banyak data yang tidak diketahui sebelumnya. *Data mining* sering dihubungkan dengan analisa simpanan data di dalam sebuah *warehouse*. Secara umum ada tiga teknik utama *Data mining* yaitu regresi, klasifikasi dan *clustering*. (Suyanti, 2017 : 8). Ada empat tahap yang dilalui dalam *data mining*, antara lain :

1. Tahap pertama : pernyataan tepat terhadap permasalahan (*Precise statement of the problem*)

Sebelum mengakses perangkat lunak *data mining*, seorang analis harus memiliki kejelasan perihal ‘pertanyaan apa yang akan ingin dijawabnya’. Jika tidak ada formula yang tepat untuk problematika yang ada maka anda hanya akan membuang-buang dan uang dalam membuat solusinya.

2. Tahap kedua : *Initial exploration*

Tahap ini dimulai dengan mempersiapkan data yang juga juga termasuk kedalam *data mining “cleaning”* (misalnya : mengidentifikasi dan menyikatkan data yang dikodekkan salah), transformasi data, memilih *subset record, data set*, langkah awal seleksi. Mendeskripsikan dan memvisualisasikan data adalah kunci dari tahap ini.

3. Tahap tiga : *Model building and validation*

Tahap ini melibatkan pertimbangan terhadap ragam permodelan dan memilih yang terbaik bagi performansi prediktif.

4. Tahap ke-empat : *Deployment*

Memilih aplikasi yang tepat berikut permodelan untuk membuat (*generate*) prediksi. Selanjutnya kita kan melihat rincian perihal tahapan-tahapan *data mining*. (Ikhwan, dkk, 2015 : 214).

II.2.1.1. Kategori Data Mining

Secara garis besar, data mining dapat dikelompokkan menjadi 2 kategori utama, yaitu :

1. *Descriptive mining*, yaitu proses untuk menemukan karakteristik penting dari data dalam satu basis data. Teknik data mining yang termasuk descriptive mining adalah clustering, asosiasi, dan sequential mining.
2. *Predictive*, yaitu proses untuk menemukan pola dari data dengan menggunakan beberapa *variable* lain di masa depan. Salah satu teknik yang terdapat dalam *predictive mining* adalah klasifikasi. (Metisen dan Sari, 2015 : 111).

II.2.1.2. Tujuan Dari Data Mining

Tujuan dari data mining yaitu :

1. *Explonatory*, yaitu untuk menjelaskan beberapa kegiatan opservasi atau kondisi.
2. *Confirmatory*, yaitu untuk mengkonfirmasi suatu hipotesis yang telah ada.
3. *Exploratory*, yaitu untuk menganalisis data baru suatu relasi yang janggal.
(Metisen dan Sari, 2015 : 111).

II.2.1.3. Langkah-Langkah Penting Data Mining

Langkah-langkah penting dalam proses penambangan pengetahuan dari data secara umum sebagai berikut :

1. Pembersihan data (*data cleaning*), yaitu proses menghapus data pengganggu (*noise*) yang dikatakan tidak konsisten atau tidak diperlukan.
2. Integrasi data (*data integration*), yaitu menggabungkan berbagai sumber data.

3. Pemilihan data (*data selection*), yaitu memilih data yang dipilih sesuai kebutuhan analisis.
4. Transformasi data (*data transformation*), yaitu proses transformasi data ke dalam format untuk diproses dan siap ditambang.
5. Penggalian data (*data mining*), yaitu menerapkan metode kecerdasan untuk ekstraksi pola.
6. Evaluasi pola (*pattern evaluation*), yaitu mengidentifikasi pola-pola yang menarik yang merepresentasikan pengetahuan.
7. Penyajian pola (*knowledge presentation*), yaitu teknik untuk memvisualisasikan pola pengetahuan ke pengguna. (Abdillah, 2016 : 499).

II.2.1.4. Operasi Dari Data Mining

Operasi data mining menurut sifatnya dibedakan menjadi dua, yaitu :

1. Prediksi (*prediction driven*) untuk menjawab pertanyaan apa dan sesuatu yang bersifat remang-remang atau transparan. Operasi prediksi digunakan untuk validasi hipotesis, *querying* dan pelaporan (misal : *spreadsheet dan pivot table*), analisis multidimensi (*dimensional summary*); OLAP (*Online Analytic Processing*) serta analisis statistik.
2. Penemuan (*discovery driven*) bersifat transparan dan menjawab pertanyaan “mengapa?”. (Halim, 2017 : 128).

II.2.2. K-Means

Metode *K-Means* adalah metode non hierarki yang paling banyak digunakan dalam pengklasteran. Algoritma K-means mudah diimplementasikan.

Pada metode ini, peneliti menentukan sendiri jumlah kluster yang akan dibentuk. Peneliti mengelompokkan entitas ke dalam k-kelompok, biasanya dilakukan secara acak. Pada masing-masing kelompok dihitung rata-ratanya. Hitung jarak setiap entitas terhadap pusat masing-masing kelompok (rata-rata kelompoknya). Masing-masing objek dialokasikan ke kluster terdekat dengan pusatnya. *Update* keanggotaan setiap entitas berdasarkan jarak terdekat dengan pusat kelompok dan ditentukan kembali pusat kluster yang baru. Proses pengalokasian objek kembali dilakukan. Suatu objek dapat berpindah ke kluster lain bila objek tersebut lebih dekat ke pusat kluster tersebut. Proses ini dilakukan secara berulang sampai tidak ada lagi entitas yang berpindah kelompok.. (Silvi, 2018 : 25).

K-means merupakan salah satu teknik pengelompokan yang bekerja berdasarkan *partitioned clustering*. Prinsip kerja dari pengelompokan *hierarchical clustering* dilakukan secara bertahap. Dan disetiap iterasi dari pengelompokan *hierarchical clustering* hanya ada satu pemilihan penggabungan suatu item terhadap item lainnya. Sedangkan prinsip kerja dari pengelompokan *partitioned clustering* adalah mengelompokkan *item* secara acak karena dipengaruhi *centroid*. Dan disetiap iterasi dari pengelompokan *partitioned clustering* dapat memungkinkan untuk terjadinya lebih dari satu pemilihan item yang akan digabungkan. (Siyanto, 2017 : 28).

Algoritma metode K-Means selanjutnya akan melakukan pengulangan langkah-langkah berikut sampai terjadi kestabilan (tidak ada objek yang dapat dipindahkan) :

1. menentukan koordinat titik tengah setiap *cluster*,
2. menentukan jarak setiap obyek terhadap koordinat titik tengah,
3. mengelompokkan obyek-obyek tersebut berdasarkan pada jarak minimumnya.

II.2.3. Clustering

Clustering (pengelompokan data) mempertimbangkan sebuah pendekatan penting untuk mencari kesamaan dalam data dan menempatkan data yang sama ke dalam kelompok-kelompok. Gagasan mengenai pengelompokan data, atau *clustering*, memiliki sifat yang sederhana dan dekat dengan cara berpikir manusia kapanpun di presentasikan jumlah data yang besar, kita biasanya cenderung merangkumkan jumlah data yang besar ini ke dalam sejumlah kecil kelompok-kelompok atau kategori-kategori untuk memfasilitasi analisisnya lebih lanjut. Bahkan, algoritma *clustering* berusaha mensegmentasi seluruh kumpulan data ke dalam subkelompok-subkelompok atau *cluster-cluster* homogen secara relatif. Dimana kesamaan *record* dalam *cluster* di maksimalkan dalam kesamaan dengan *record* diluar *cluster* ini diminimalkan. (Syahputra, dkk, 2018 : 2).

Clustering merupakan bagian dari ilmu *Data mining* yang bersifat tanpa arahan (*unsupervised*). *Clustering* adalah proses pembagian data ke dalam kelas atau cluster berdasarkan tingkat kesamaannya. *Clustering* merupakan pekerjaan yang memisahkan data atau vektor ke dalam sejumlah kelompok atau *cluster* menurut karakteristiknya masing-masing. Data-data yang memiliki kemiripan karakteristik akan berkumpul dalam kelompok atau *cluster* yang sama. Data-data yang memiliki perbedaan karakteristik, akan berkumpul dalam kelompok atau

cluster yang berbeda. Tujuan utama dari metode *clustering* adalah pengelompokan sejumlah data atau obyek ke dalam *cluster (group)* sehingga dalam setiap *cluster* akan berisi data yang semirip mungkin. (Siyamto, 2017 : 28).

Langkah-langkah pada proses *clustering* dengan menggunakan algoritma *K-Means*, adalah sebagai berikut :

1. Tentukan jumlah *cluster (k)* pada *data set*
2. Tentukan nilai pusat (*centroid*)

Penentuan nilai *centroid* pada tahap awal dilakukan secara random dengan rumus menentukan target awal *k-means*, rumus tersebut digunakan untuk mendapatkan target data atau jarak antara kelompok, yaitu titik pusat awal untuk menghitung algoritma *k-means* iterasi 0 seperti pada persamaan berikut ini :

Jumlah Data/Jumlah Class + 1

Keterangan :

Jumlah data = Jumlah data yang akan digunakan

Jumlah *class* = Jumlah kelompok yang telah ditentukan sebelumnya seperti sangat tinggi, tinggi, normal, rendah, sangat rendah.

Sedangkan pada tahap *iterasi* digunakan rumus rata-rata, pada rumus tersebut dilakukan perhitungan untuk mencari nilai rata-rata seperti pada persamaan berikut ini :

$$V_{ij} = \frac{1}{N_i} \sum_{k=0}^{N_i} X_{kj}$$

Keterangan :

V_{ij} = *centroid* rata-rata *cluster* ke-i untuk variabel ke-j

N_i = jumlah anggota *cluster* ke-i

i, k = indeks dari *cluster*

j = indeks dari variabel

X_{kj} = nilai data ke- k variabel ke- j dalam *cluster* tersebut.

3. Pada masing-masing record, hitung jarak terdekat dengan *centroid*. Jarak *centroid* yang digunakan adalah *Euclidean Distance*, dengan rumus seperti pada persamaan berikut :

$$De = \text{Sqrt}((x_i - s_i)^2 + (y_i - t_i)^2)$$

Keterangan :

De = *Euclidean Distance*

i = banyaknya objek

(x, y) = koordinat objek

(s, t) = koordinat *centroid*

4. Kelompokkan objek berdasarkan jarak ke *centroid* terdekat

5. Ulangi langkah ke-2, lakukan *iterasi* hingga *centroid* bernilai optimal.

(Abdillah, dkk, 2016 : 500).

II.2.4. Customer

Pelanggan (*Customer*) adalah semua orang yang menuntut perusahaan kita untuk memenuhi standard kualitas tertentu, dan karena itu akan memberikan pengaruh pada performansi (*performance*) perusahaan. (Wahyuni dan Amboningtyas, 2017 : 2).

II.2.5. Visual Basic 2010

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic* 2010. Beberapa pengembangan yang terdapat

didalamnya antara lain dukungan terhadap *library* terbaru dari *Microsoft*, yaitu *.Net Framework* 4.0, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap *database-database*, baik *standalone* maupun *database server*. (Sari, dkk, 2010 : 2).

II.2.6. Normalisasi

Normalisasi Normalisasi merupakan cara pendekatan dalam membangun desain lojik basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan tabel yang normal. Selain itu normalisasi merupakan proses pendekatan formal untuk menelaah dan mengelompokkan item/field/atribut ke bentuk yang lebih baik dalam menghadapi perubahan-perubahan di masa mendatang serta meminimumkan pengaruh perubahan-perubahan pada sistem aplikasi program. (Wahyudi, 2015 : 47).

II.2.7. Database

Database merupakan kumpulan *file-file* yang saling berkaitan dan berinteraksi, relasi tersebut bila ditunjukkan dengan kunci dari tiap-tiap *file* yang ada. Satu database menunjukkan suatu kumpulan data yang dipakai dalam suatu lingkup perusahaan, instansi. Pengolahan *database* merupakan suatu cara yang dilakukan terhadap *file-file* yang berada di suatu instansi yang mana *file* tersebut dapat disusun, diurut, diambil sewaktu-waktu serta dapat ditampilkan dalam bentuk suatu laporan sehingga dapat mengolah *file-file* yang berisikan informasi tersebut secara rapi. (Sovia dan Febio, 2015 : 41).

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi. *Database* mempunyai kegunaan dalam mengatasi penyusunan dan penyimpanan data, maka seringkali masalah yang dihadapi adalah :

1. Redundansi dan Inkonsistensi data.
2. Kesulitan dalam pengaksesan data.
3. Isolasi data untuk standarisasi.
4. *Multi user*.
5. Keamanan data.
6. Integritas data.
7. Kebebasan data. (Urva dan Siregar, 2015 : 93).

II.2.8. SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. SQL Server adalah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. (Widya dan Zulkarnaen, 2015 : 3).

II.2.9. *Unified Modeling Language (UML)*

Menurut Windu Gata (2015) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.


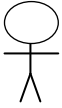

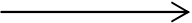
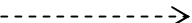
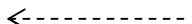
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Urva dan Siregar, 2015 : 93).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram 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. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol Use Case


Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki <i>control</i> terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.



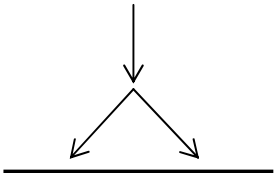
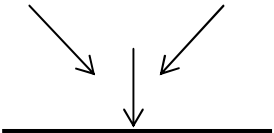
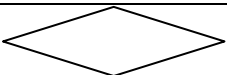
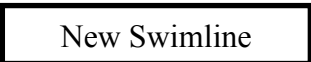
(Sumber : Urva dan Siregar, 2015 : 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.

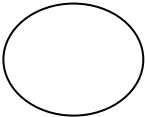
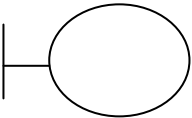
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.

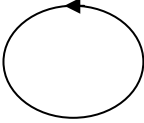
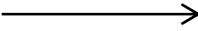
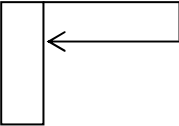


(Sumber : Urva dan Siregar, 2015 : 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.

	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang di kirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Urva dan Siregar, 2015 : 95)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Urva dan Siregar, 2015 : 95)