

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu :

1. Anindya Khrisna Wardhani, 2016, dengan judul “Implementasi algoritma K-Means untuk pengelompokkan penyakit pasien pada puskesmas kaje pekalongan (Jurnal Transformatika, Volume 14, Nomor 1, Juli 2016)” Penelitian ini menghasilkan suatu sistem data mining yang mampu menampilkan hasil pengelompokkan penyakit pasien berdasarkan data penyakit yang ada di puskesmas kaje pekalongan.
2. Nurul Rohmawati W, 2015, dengan judul “Implementasi Algoritma K-Means Dalam Pengklasteran Mahasiswa Pelamar Beasiswa” Penelitian ini menghasilkan sebuah sistem data mining yang mampu menampilkan hasil pengelompokkan pelamar beasiswa bantuan belajar mahasiswa.
3. Yulia Darni, 2016, dengan judul “Penerapan Metode Clustering K-Means Dalam Pengelompokkan Penjualan Produk” Penelitian ini menghasilkan sebuah sistem data mining yang mampu menampilkan hasil dari pengelompokkan penjualan produk. Dengan tujuan untuk membantu pihak Minimarket Tika untuk menentukan Produk apa saja yang laku dan tidak

laku dari banyanya produk yang ada di minimarket tersebut tanpa harus memakan waktu lama karena penilaiannya yang dilakukan objektif dengan menerapkan metode K-Means.

4. Nurhikmah, 2018, dengan judul “Penerapan Data Mining Menggunakan Algoritma K-Means Dalam Pengelompokan Data Peminjaman Buku Di Perpustakaan Untuk Mengetahui Minat Belajar Siswa (Studi Kasus : Yayasan Perguruan Al-Fattah Medan)” Penelitian ini menghasilkan aplikasi data mining yang mampu mengelompokkan data peminjaman buku di Yayasan Perguruan Al-Fattah Medan dengan menggunakan metode K-Means dengan tujuan untuk mengetahui minat belajar siswa. Penelitian dirancang dan dibangun menggunakan *software Microsoft Visual Basic 2010* dan *database SQL Server 2008*. Hasil akhir dari penelitian ini adalah berupa laporan perhitungan metode K-Means.

## **II.2. Landasan Teoritis**

### **II.2.1. Data Mining**

Data mining adalah Data mining adalah analisa terhadap data untuk menemukan hubungan yang jelas serta menyimpulkannya yang belum diketahui sebelumnya dengan cara terkini dipahami dan berguna bagi pemilik data tersebut (Yulia Darni, 2016).

Data mining yang juga dikenal dengan istilah *Pattern Recognition* merupakan suatu metode yang digunakan untuk pengolahan data guna menemukan pola yang tersembunyi dari data yang diolah. Data yang diolah

dengan teknik data mining ini kemudian menghasilkan suatu pengetahuan baru yang bersumber dari data lama, hasil dari pengolahan data tersebut dapat digunakan dalam menentukan keputusan di masa depan. Data mining juga bisa diartikan sebagai rangkaian kegiatan untuk menemukan pola yang menarik dari data dalam jumlah besar, kemudian data– data tersebut dapat disimpan dalam *Database*, data *Warehouse* atau penyimpanan informasi (Anindya Khrisna Wardhani, 2016).

Menurut Fina Nasari, Surya Darma (2015), 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.

Data mining merupakan disiplin ilmu yang mempelajari metode untuk mengekstrak pengetahuan atau menemukan pola dari suatu data. Data mining adalah suatu metode pengolahan data untuk menemukan pola yang tersembunyi dari data tersebut. Hasil dari pengolahan data dengan metode data mining ini dapat digunakan untuk mengambil keputusan di masa depan (Heni Sulastri, dkk 2017).

Data mining adalah proses menemukan hubungan baru yang mempunyai arti, pola dan kebiasaan dengan memilah-milah sebagian besar data yang disimpan dalam media penyimpanan dengan menggunakan teknologi pengenalan pola seperti teknik statistik dan matematika. Data mining merupakan gabungan dari beberapa disiplin ilmu yang menyatukan teknik dari mesin pembelajaran, pengenalan pola, statistik dan *database* (Yuli Mardi, 2016).

## II.2.2. Clustering

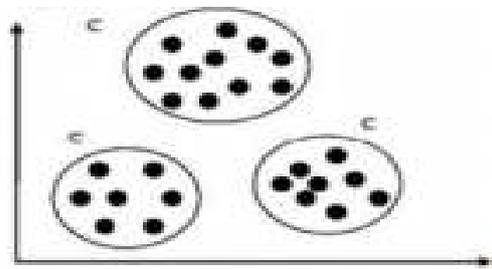
*Clustering* atau pengklasteran adalah suatu teknik data mining yang digunakan untuk menganalisis data untuk memecahkan permasalahan dalam pengelompokkan data atau lebih tepatnya mempartisi dari dataset ke dalam subset. Pada teknik *clustering* targetnya adalah untuk kasus pendistribusian (objek, orang, peristiwa dan lainnya) ke dalam suatu kelompok, hingga derajat tingkat keterhubungan antar anggota *cluster* yang sama adalah kuat dan lemah antara anggota *cluster* yang berbeda (Anindya Khrisna Wardhani, 2016).

Teknik *Cluster* terbagi dua yaitu *Hierarchical Clustering* dan *Non-Hierarchical Clustering*. *Hierarchical Clustering* merupakan suatu metode pengelompokkan data yang cara kerjanya dengan mengelompokkan dua data atau lebih yang mempunyai kesamaan atau kemiripan, kemudian proses dilanjutkan ke objek lain yang memiliki kedekatan dua, proses ini terus berlangsung hingga *Cluster* membentuk semacam *Tree* dimana ada hirarki atau tingkatan yang jelas antar objek dari yang paling mirip hingga yang paling tidak mirip. Namun secara logika semua objek pada akhirnya hanya akan membentuk sebuah *cluster* (Anindya Khrisna Wardhani, 2016).

*Non-Hierarchical Clustering* merupakan teknik yang dimulai dengan menentukan jumlah *Cluster* yang diinginkan (Dua *Cluster*, Tiga *Cluster*, Empat *Cluster* atau lebih), setelah jumlah yang *cluster* yang diinginkan maka proses *cluster* dimulai tanpa mengikuti proses hirarki, metode ini juga sering *cluster* dimulai tanpa mengikuti proses hirarki, metode ini juga sering disebut sebagai metode K-Means *Clustering*. Pada proses analisis *cluster* metode yang digunakan

untuk membagi data menjadi subset data berdasarkan kesamaan atau kemiripan yang telah ditentukan sebelumnya. Jadi analisis *cluster* secara umum dapat dikatakan bahwa (Anindya Khrisna Wardhani, 2016) :

- a. Data yang terdapat dalam satu *Cluster* memiliki tingkat kesamaan yang tinggi.
- b. Dan yang terdapat dalam suatu *Cluster* yang berbeda memiliki tingkat kesamaan yang rendah.



**Gambar II.1. Grafik *Clustering***  
(Sumber : Anindya Khrisna Wardhani ; 2016)

Pada penelitian ini teknik *Clustering* yang digunakan adalah teknik *Non-Hierarchical Clustering*.

Untuk masalah pengelompokkan data berdasarkan kemiripan atau ketidakmiripan antar-data tanpa ada label kelas yang diketahui sebelumnya disebut dengan pembelajaran tidak terbimbing atau *unsupervised learning*. Dalam konteks yang lain, pembelajaran tidak terbimbing disebut juga pengelompokkan atau *clustering* (Eko Prasetyo, 2014 ; 13).

### **II.2.3. K-Means**

K-Means merupakan suatu algoritma yang digunakan dalam pengelompokkan secara pertisi yang memisahkan data ke dalam kelompok yang berbeda-beda. Algoritma ini mampu meminimalkan jarak antara data ke

clusternya. Pada dasarnya penggunaan algoritma ini dalam proses *clustering* tergantung pada data yang didapatkan dan konklusi yang ingin dicapai di akhir proses (Anindya Khrisna Wardhani, 2016).

Algoritma K-Means mengelompokkan data berdasarkan titik pusat *Cluster* (*Centroid*) terdekat dengan data. Tujuan dari K-Means adalah pengelompokkan data dengan memaksimalkan kemiripan data dalam satu *Cluster* dan meminimalkan kemiripan data antar *Cluster*. Sehingga pemaksimalan kemiripan data didapatkan berdasarkan jarak terpendek antara data terhadap titik *Centroid* (Asroni, Ronald Adrian, 2015).

Pada dasarnya penggunaan algoritma K-Means dalam melakukan proses *Clustering* tergantung pada data yang ada dan konklusi yang ingin dicapai. Untuk itu digunakan algoritma K-Means yang didalamnya terdapat aturan sebagai berikut (Fenty Eka M. Agustin, dkk, 2015) :

- a. Jumlah *Cluster* yang perlu diinputkan.
- b. Hanya memiliki atribut bertipe numerik.

Algoritma K-Means merupakan algoritma pengelompokkan iteratif yang melakukan partisi set data ke dalam sejumlah *K cluster* yang sudah di tetapkan di awal. Algoritma K-Means sederhana untuk diimplementasikan dan dijalankan, relatif cepat, mudah beradaptasi, umum penggunaannya dalam praktek. Secara historis, K-Means menjadi salah satu algoritma yang paling penting dalam bidang data mining (Eko Prasetyo, 2014 ; 189).

Adapun tahapan-tahapan yang ada dalam proses algoritma K-Means untuk *Clustering* yaitu sebagai berikut :

1. Tentukan  $k$  sebagai jumlah *Cluster* yang ingin di bentuk. Tetapkan pusat *Cluster* (*Centroid*).
2. Hitung jarak setiap data ke pusat *Cluster* menggunakan persamaan *Euclidean*.
3. Kelompokkan data ke dalam *Cluster* dengan jarak yang paling dekat dengan pusat *Cluster*.
4. Hitung pusat *Cluster* yang baru.
5. Ulangi langkah ke 2 sampai dengan langkah yang ke 4 data yang berpindah ke *cluster* yang lain (Fenty Eka M. Agustin, dkk, 2015).

Rumus yang digunakan dalam melakukan proses *Clustering* dengan algoritma K-Means adalah sebagai berikut :

Untuk menghitung jarak setiap data ke pusat *cluster* menggunakan persamaan *Euclidean* :

$$d_{ik} = \sqrt{\sum_j^m (C_{ij} - C_{jk})^2} \dots\dots\dots(1)$$

dimana :

$d_{ik}$  = jarak setiap data ke  $i$  ke pusat *cluster*  $j$

$C_{ij}$  = data ke  $i$  pada atribut data ke  $j$

$C_{jk}$  = titik pusat ke  $j$  pada atribut ke  $k$

Untuk menghitung pusat *cluster* yang baru :

$$C_{jk} = \frac{\sum_{i=1}^P X_{ij}}{P} \dots\dots\dots(2)$$

dimana :

$C_{jk}$  = *centroid* data

$X_{ij}$  = anggota data yang termasuk ke dalam *centroid* tertentu

P = jumlah data yang menjadi anggota *centroid* tertentu (Fenty Eka M. Agustin, dkk, 2015).

#### II.2.4. *Microsoft Visual Studio 2010*

*Visual Studio 2010* pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual Studio 2010* (yang sering juga disebut dengan VB .Net 2010) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows. *Visual basic* adalah sebuah bahasa pemrograman yang berpusat pada *object (Object Oriented Programming)* digunakan dalam pembuatan aplikasi *Windows* yang berbasis *Graphical User Interface*, hal ini menjadikan *Visual Basic* menjadi bahasa pemrograman yang wajib diketahui dan dikuasai oleh setiap programmer. Beberapa karakteristik obyek tidak dapat dilakukan oleh *Visual Basic* misalnya seperti *Inheritance* tidak bisa module dan *Polymorphism* secara terbatas bisa dilakukan dengan deklarasi *class module* yang mempunyai *Interface* tertentu (Ninuk Wiliani, 2017 : 78)

### **II.2.5. Database**

Data adalah fakta yang dapat direkam dan memiliki arti secara implisit. Sedangkan kumpulan data yang memiliki hubungan secara implisit itu disebut *Database*. Basis data merupakan kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan. Dalam *database* juga dikenal istilah DBMS (*Database Management Systems*) yaitu sekumpulan program yang memungkinkan pengguna untuk membuat dan memelihara suatu *database*. Bisa juga dikatakan bahwa DBMS merupakan perangkat *General Purpose Software System* yang berfungsi untuk mewartahi proses-proses dalam *database* seperti pendefinisian, pembuatan, *sharing*, maupun manipulasi *database* (Fitri Marisa, 2015).

### **II.2.6. SQL Server 2008**

*SQL (Structure Query Language)* merupakan bahasa *query* standar yang dipergunakan untuk mengakses basis data relasional. *Statement SQL* secara garis besar dibagi menjadi 3 kategori yaitu *Data Definition Languages (DDL)*, *Data Manipulation Languages (DML)* dan *Data Control Language (DCL)*. Dalam *SQL Server* juga dikenal dengan istilah RDBMS (*Relational Database Management System*). RDBMS adalah dasar untuk *SQL*, dan untuk semua sistem *database modern* seperti *Microsoft SQL Server*, *IBM DB2*, *Oracle*, *MySQL*, dan *Microsoft Access*. Data dalam RDBMS disimpan dalam objek *database* yang disebut tabel (Fitri Marisa, 2015).

### II.2.7. *Unified Modeling language (UML)*

*Unified Modeling Language (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 (Ade Hendini, 2016).

Adapun jenis-jenis yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

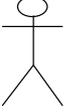
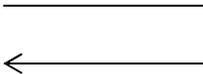
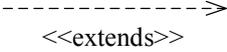
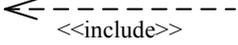
a. *Use Case Diagram*

*Use Case Diagram* merupakan pemodelan untuk kelakuan (*Behavior*) sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2016).

Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat dari Tabel II.1 dibawah ini :

**Tabel II.1. Simbol *Use Case Diagram***

Gambar	Nama	Keterangan
	<i>Use Case</i>	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.

	<i>Actor</i>	<i>Actor</i> 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.
	Asosiasi	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 mengindikasikan data.
	<i>Extends</i>	<i>Extends</i> merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.
	<i>Include</i>	<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.

(Sumber : Ade Hendini ; 2016)

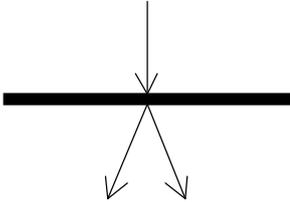
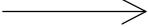
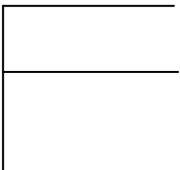
#### b. *Activity Diagram*

*Activity diagram* adalah diagram yang menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang

dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktivitas-aktivitas tersebut bergantung satu sama lain. *Activity Diagram* menggambarkan *Work Flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Ade Hendini, 2016).

Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat dari Tabel II.2 dibawah ini :

**Tabel II.2. Simbol *Activity Diagram***

Gambar	Nama	Keterangan
	<i>Initial State / Start State</i>	<i>Start Point</i> diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>Action State</i>	menggambarkan suatu proses atau kegiatan bisnis.
	<i>Transition</i>	digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Control Flow</i>	digunakan untuk menunjukkan kegiatan.
	<i>Decision</i>	Menggambarkan pilihan untuk pengambilan keputusan <i>true</i> atau <i>false</i> .
	<i>Final State</i>	Digunakan untuk mengakhiri aktivitas.
	<i>Swimlane</i>	Merupakan pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa.

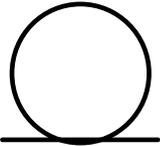
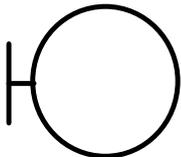
(Sumber : Ade Hendini ; 2016)

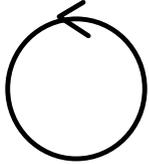
c. *Sequence Diagram*

*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Ade Hendini, 2016).

Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu :

**Tabel II.3. Simbol *Sequence Diagram***

Gambar	Nama	Keterangan
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i>	garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .
	<i>Message</i>	simbol mengirim pesan antar <i>class</i> .
	<i>Entity Class</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>	Merupakan berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>

	<i>Control class</i>	Merupakan suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
-----------------------------------------------------------------------------------	----------------------	------------------------------------------------------------------------------------------------------

(Sumber : Ade Hendini ; 2016)

#### d. *Class Diagram*

*Class Diagram* 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 (Ade Hendini, 2016).

*Class Diagram* secara khas meliputi : Kelas (*Class*), *Relasi Assosiations*, *Generalitation* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operation / Method*) dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016).

**Tabel II.4. *Multiplicity Class Diagram***

<i>Multiplicity</i>	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 maksimal 4.
--	----------------------------

**(Sumber : Ade Hendini ; 2016)**