

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Sebelumnya**

Berdasarkan penelitian yang dilakukan oleh Sembiring, dkk (2018) mengenai Analisa Kinerja Algoritma C.45 Dalam Memprediksi Hasil Belajar, Sembiring, dkk menganalisa kinerja algoritma C45 dalam memprediksi hasil belajar, sedangkan penelitian ini untuk memprediksi pola pembelian bahan campuran olahan karet.

Berdasarkan penelitian yang dilakukan oleh Saleh (2017) mengenai Prediksi Kebangkrutan Perusahaan Menggunakan Algoritma C4.5 Berbasis *Forward Selection*, Shaleh menggunakan metode algoritma C45 untuk memprediksi kebangkrutan suatu perusahaan, sedangkan penelitian ini adalah untuk memprediksi pola pembelian bahan campuran olahan karet.

Berdasarkan penelitian yang dilakukan oleh Elisa (2018) mengenai Prediksi Profit Pada Perusahaan Dengan Klasifikasi Algoritma C4.5, Elisa menggunakan metode algoritma C45 untuk memprediksi profit perusahaan, sedangkan penelitian ini adalah untuk memprediksi pola pembelian bahan campuran olahan karet.

Berdasarkan penelitian yang dilakukan oleh Sundari (2019) mengenai Implementasi *Data Mining* Dengan Algoritma C45, Sundari menggunakan metode algoritma C45 untuk memprediksi pembelian tipe sepeda motor, sedangkan penelitian ini adalah untuk memprediksi pola pembelian bahan campuran olahan karet.

## II.2. Landasan Teori

Berikut ini adalah beberapa landasan teori yang digunakan dalam pembuatan proposal ini :

### II.2.1. *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, dan kecerdasan buatan untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar. (Handiwidjojo dan Sukendro, 2015 : 152).

Masukan dari proses *knowledge discovery in database* adalah data mentah yang didalamnya terdapat informasi penting atau dibutuhkan oleh penggali informasi. Diperlukan proses pengolahan data mentah sebelum masuk dalam proses data mining. Tahap pengolahan data mentah ini berada dalam tahap praproses. Tahap ini digunakan untuk mengubah data menjadi format yang ditentukan dan dibutuhkan dalam proses data mining. Tahap ini juga termasuk dalam penggabungan data dari berbagai sumber menjadi satu, penghapusan *noise* dan data duplikat, dan pemilihan atribut yang dibutuhkan dan relevan. Tahap ini merupakan tahap yang menghabiskan banyak sumber daya, tetapi tahap ini merupakan tahapan penting dalam keseluruhan proses *knowledge discovery in database*. Hal ini disebabkan kualitas data masukan memberikan pengaruh terhadap hasil keluaran proses penggalian data. Tahap data mining dilaksanakan setelah tahap pra-proses data. Penggalian data dilakukan dengan observasi pola

dan hubungan dalam data menggunakan algoritma *machine learning*. Pola dan hubungan ini akan menghasilkan informasi yang berguna. Tahap terakhir adalah *post-processing*, dimana dalam tahap ini hasil penggalian data akan diintegrasikan dengan sistem pengambilan keputusan atau divisualisasikan dalam bentuk memudahkan pengguna dalam pembacaan informasi. (Arifiyanti, dkk, 2018 : 81).

#### **II.2.1.1. Jenis *Data Mining***

*Data mining* secara umum terbagi menjadi dua jenis yaitu sebagai berikut:

1. Prediktif. Tujuan metode dalam jenis ini adalah untuk memprediksi nilai dari target atribut berdasarkan nilai dari atribut-atribut lainnya. Klasifikasi dan regresi termasuk ke dalam jenis prediktif.
2. Deskriptif. Tujuan dari metode *data mining* ini adalah untuk mendapatkan pola yang menghasilkan simpulan hubungan dari data. Metode ini digunakan untuk menjelaskan pola dan hubungan data. *Clustering* termasuk dalam metode deskriptif. (Arifiyanti, dkk, 2018 : 81).

#### **II.2.1.2. Pohon Keputusan**

Pohon keputusan menggunakan representasi struktur pohon (*tree*) di mana setiap *node* merepresentasikan atribut, cabangnya merepresentasikan nilai dari atribut dan daun merepresentasikan kelas. *Node* yang paling atas dari pohon keputusan disebut sebagai *root*. Salah satu keuntungan yang paling signifikan dari pohon keputusan adalah kenyataan bahwa pengetahuan dapat diekstrak dan direpresentasikan dalam bentuk klasifikasi aturan if-then. Pohon keputusan merupakan metode klasifikasi yang paling populer digunakan. Selain karena

pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami. Pohon keputusan bekerja dengan membentuk pohon keputusan yang dapat disimpulkan aturan-aturan klasifikasi tertentu, salah satunya adalah C4.5. (Azwanti, 2018 : 2).

### II.2.2. Algoritma C45

Algoritma C45 adalah salah satu metode untuk membuat *decision tree* berdasarkan *training* data yang telah disediakan. Secara umum algoritma C45 untuk membangun pohon keputusan adalah sebagai berikut :

1. Pilih atribut sebagai akar.
2. Buat cabang untuk masing-masing nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk masing-masing cabang sampai semua kasus selesai.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *Gain* tertinggi dari atribut-atribut yang ada. Untuk memilih atribut sebagai akar, didasarkan pada nilai *Gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *Gain* digunakan rumus seperti tertera dalam Rumus II.1. Rumus algoritma C45 adalah :

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^n \frac{S_i}{S} * Entropy(S_i)$$

Keterangan:

S : Himpunan Kasus

A : Atribut

n : Jumlah Partisi Atribut A

$S_i$  : Jumlah Kasus Pada Partisi ke-i

S : Jumlah Kasus dalam S.

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_z p_i$$

Keterangan:

S : Himpunan Kasus

A : Fitur

n : Jumlah partisi S

$p_i$  : Proporsi dari  $S_i$  terhadap S. (Pramadani, dkk, 2018 : 246).

### II.2.3. *Hypertext Preprocessor (PHP)*

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada *server* sedang yang dikirim ke *browser* hanya hasilnya saja. Ketika seorang pengguna *internet* membuka situs yang menggunakan fasilitas *server side scripting* PHP, maka terlebih dahulu *server* yang bersangkutan akan memproses semua perintah PHP di *server* lalu mengirimkan hasilnya dalam format HTML ke *web server* pengguna *internet* tadi. Sehingga kode asli yang ditulis dengan PHP tidak terlihat di *browser* pengguna. PHP berfungsi mengambil informasi dari *form* berbasis *web* dan menggunakannya untuk berbagai macam fungsi, sebagai bahasa untuk mengidentifikasi seberapa banyak pengunjung menggunakan bahasa PHP, Pengaturan *layout* dalam berbagai macam *browser* seperti *Firefox*, Bahasa Pemrograman PHP dalam *web* sangat luas, jadi semua tergantung pada anda, kalau anda sangat teliti anda akan tahu fungsinya. (Wulandari dan Aprilia, 2015 : 43).

#### **II.2.4. MySQL**

MySQL adalah suatu perangkat lunak *database* relasi atau *Relational Database Management System* (RDBMS) yang didistribusikan gratis di bawah lisensi GPL (*General Public License*). Dimana setiap orang bebas menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang dijadikan *closed source* atau komersial. (Muslihudin dan Larasati, 2014 : 34).

#### **II.2.5. Prediksi**

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat di perkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi. (Herdianto, 2013 : 8).

#### **II.2.6. Karet**

Karet alam merupakan suatu komoditi pertanian yang penting baik untuk lingkup internasional dan terutama di Indonesia. Di Indonesia karet merupakan suatu hasil pertanian terkemuka karena banyak menunjang perekonomian Negara. Sampai tahun 1992 ada tiga Negara yang menguasai pasaran karet dunia yaitu Indonesia, Thailand dan Malaysia. (Cut Fatimah Zahra, 2006 : Karet)

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

*Unified Modeling Language* (UML) adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun


perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek. UML saat ini sangat banyak digunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. (Urva dan Siregar, 2015 : 95).

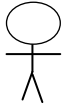


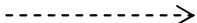
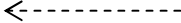
Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML adalah sebagai berikut :

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
	<p><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>.</p>




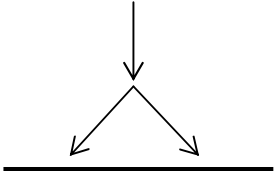
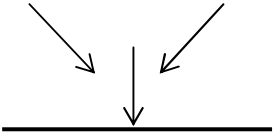
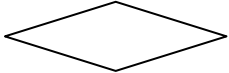
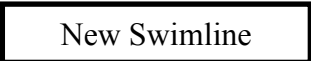
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.</p> <p>Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.</p> <p>Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki <i>control</i> terhadap <i>use case</i>.</p>
	<p>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 aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><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.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(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***

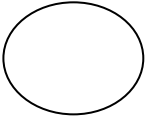
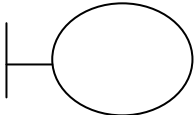
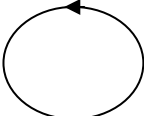

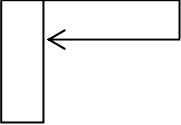
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork (Percabangan)</i> , digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join (penggabungan) atau rake</i> , 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.

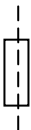

(Sumber : Urva dan Siregar, 2015 : 94)

### 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>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> , 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 dikirim untuk dirinya sendiri.

	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(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***

<b>Multiplicity</b>	<b>Penjelasan</b>
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)**