

BAB II

LANDASAN TEORI

2.1 Kajian Terhadap Penelitian Yang Terkait Sebelumnya

Dalam penelitian (Wahyuni, 2020) yang bertujuan memprediksi harga jual cabai menggunakan algoritma k-nearest neighbor. Dalam penelitian menggunakan data set yang sudah ada sebelumnya.

Dalam jurnal ilmiahnya (Palar et al., 2016) membahas faktor-faktor yang mempengaruhi harga cabai rawit di kota Manado. Untuk mengetahui jumlah permintaan pada cabai, perlu diketahui dahulu jumlah populasi. Setelah mengetahui jumlah penduduk, selanjutnya menentukan data sampel.

Dalam jurnal (Zhai et al., 2015) mengkaji tentang pengklasifikasian data untuk menentukan hasil produksi kelapa sawit dengan perhitungan euclidean.

2.2 Metode K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek yang berdasarkan dari data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN merupakan algoritma supervised learning dimana hasil dari query instance yang baru diklasifikan berdasarkan mayoritas dari kategori pada algoritma KNN. Dimana kelas yang paling banyak muncul yang nantinya akan menjadi kelas hasil dari klasifikasi [4].

Pada jurnal (Prasetya & Ridwan, 2019) Algoritma K – Nearest Neighbor atau (KNN) adalah algoritma yang mengklasifikasikan berdasarkan kedekatan (jarak) suatu data dengan data lainnya. Dekat atau jauhnya (jarak) biasanya dihitung berdasarkan Euclidean dengan rumus sebagai berikut :

1. Menentukan nilai k (tetangga terdekat).
2. Menghitung kuadrat jarak Euclidean objek terhadap data training yang akan diberikan.
3. Selanjutnya mengurutkan hasil no 2 secara ascending.
4. Mengumpulkan kategori (klasifikasi nearest neighbour berdasarkan nilai k).
5. Dengan menggunakan kategori nearest neighbour yang paling mayoritas maka dapat diprediksi objek yang baru.

Untuk $P = (p_1, p_2, \dots, p_n)$ dan $Q = (q_1, q_2, \dots, q_n)$, maka

$$\text{Jarak} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

$$\text{Jarak} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

2.3 Metode Naïve Bayes

Naïve Bayes adalah salah satu algoritma pembelajaran induktif yang paling efektif dan efisien untuk machine learning dan data mining. Performa naïve bayes yang kompetitif dalam proses klasifikasi walaupun menggunakan asumsi keindependenan atribut (tidak ada kaitan antar atribut). Asumsi keindependenan atribut ini pada data sebenarnya jarang terjadi, namun walaupun asumsi keindependenan atribut tersebut dilanggar performa pengklasifikasian naïve bayes cukup tinggi, hal ini dibuktikan pada berbagai penelitian empiris. Naïve Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai teorema Bayes. Teorema tersebut dikombinasikan dengan "naive" dimana diasumsikan kondisi antar atribut saling bebas. Formulasi Naïve bayes untuk

klasifikasi dalam jurnal (Putri et al., 2014) Prediksi Naïve Bayes didasarkan pada teorema bayes dengan rumus klasifikasi sebagai berikut(Prasetya & Ridwan, 2019):

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^q P(X_i|Y)}{P(X)}$$

Sementara Naïve Bayes dengan fitur berkelanjutan memiliki rumus :

$$P(Y | X) = \frac{1}{\sqrt{2\pi} \sigma} \exp \frac{-(x-\mu)^2}{2 \sigma^2}$$

$P(Y|X)$ = Probabilitas data dengan vector X pada kelas Y.

$P(Y)$ = Probabilitas awal kelas Y.

$\prod_{i=1}^q P(X_i|Y)$ = Probabilitas independen kelas Y dari semua fitur dalam vector X.

μ = Nilai rata-rata atribut dengan atribut dengan fitur lanjutan

σ = simpangan baku

2.4 UML

Menurut (Nugroho,2010), pengertian UML atau Unified Modeling Language adalah bahasa pemodelan khusus untuk sistem atau perangkat lunak dengan paradigma “berorientasi objek”. Pemodelan atau modeling sebenarnya digunakan khusus untuk menyederhanakan berbagai kendala yang kompleks sehingga nantinya mudah untuk dipelajari dan dipahami.

2.4.1 Use Case Diagram

Menurut (A.S Rosa dan Salahuddin M,2011) dalam Buku“Modul Pembelajaran Rekayasa Perangkat Lunak”. *Use case* adalah Diagram yang menggambarkan apa saja aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar.

2.4.2 Activity Diagram

Menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas (Rosa dan Shalahuddin, 2014).

2.4.3 Class Diagram

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. Berikut adalah simbol-simbol yang ada pada diagram Kelas (Rosa dan Shalahuddin, 2013).

2.4.4 Entity Relationship Diagram (ERD)

Menurut (Kusrini dan Koniyo 2007) mengatakan bahwa “Entity Relationship Diagram (ERD) merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antardata, karena hal ini relatif kompleks.” Entity Relationship Diagram (ERD) menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antardata. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu (Kusrini dan Koniyo 2007).

1. Entity

Entity adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Entitas digambarkan dalam bentuk persegi empat.

2. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

3. Hubungan/Relationship

Sebagaimana halnya entitas, hubungan pun harus dibedakan antara hubungan atau bentuk hubungan antarentitas dengan isi dari hubungan itu sendiri. Relationship digambarkan dalam bentuk intan (diamonds).

2.4.5 Deployment Diagram

Deployment Diagram menggambarkan detail bagaimana komponen deploy dalam infrastruktur system, dimana komponen akan terletak (pada mesin, server atau perangkat keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.