

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Naïve Bayes

Metode Naïve Bayes merupakan metode yang digunakan memprediksi probabilitas. Sedangkan klasifikasi Bayes adalah klasifikasi statistik yang dapat memprediksi kelas suatu anggota probabilitas. Untuk klasifikasi Bayes sederhana yang lebih dikenal sebagai *Naïve Bayesian Classifier* dapat diasumsikan bahwa efek dari suatu nilai atribut sebuah kelas yang diberikan adalah bebas dari atribut-atribut lain. *Naïve Bayes Classifier* merupakan sebuah metode klasifikasi yang berakar pada teorema Bayes. Ciri utama dari *Naïve Bayes Classifier* ini adalah asumsi yang sangat kuat (naif) akan independensi dari masing-masing kondisi/kejadian (Fais dkk, 2014) dalam (Syarifudin, Hidayat, & Fanani, 2018).

Secara garis besar Teorema Bayes memiliki bentuk umum sebagai berikut (Dahri, Agus, & Khairina, 2016).

$$P(C|X) = \frac{P(X|C) \times P(C)}{P(X)} \quad (1)$$

Keterangan:

$X$  = Data dengan kelas yang belum diketahui

$C$  = *Hipotesis* data  $X$  merupakan suatu kelas spesifik

$P(C)$  = Probabilitas *hipotesis C* (probabilitas *prior*)

$P(X)$  = Probabilitas  $X$

$P(C|X)$  = Probabilitas *hipotesis C* berdasarkan kondisi  $X$  (probabilitas *posterior*)

$P(X|C)$  = Probabilitas  $X$  berdasarkan kondisi pada *hipotesis C*

Atau dengan kata lain persamaan diatas dapat digambarkan sebagai persamaan (2).

$$Posterior = \frac{Likelihood \times Prior}{Evidence} \quad (2)$$

Perhitungan naïve bayes dapat dilakukan dengan langkah berikut ini:

1. Mencari nilai *prior* untuk tiap-tiap kelas dengan menghitung rata-rata tiap kelas dengan menggunakan persamaan (3).

$$P = \frac{X}{A} \quad (3)$$

Keterangan:

$P$  = Nilai *prior*

$X$  = Jumlah data tiap kelas

$A$  = Jumlah data seluruh kelas

2. Mencari nilai *likelihood* untuk tiap-tiap kelas dengan menggunakan persamaan (4).

$$L = \frac{F}{B} \quad (4)$$

Keterangan:

$L$  = Nilai *likelihood*

$F$  = Jumlah data fitur tiap kelas

$B$  = Jumlah data seluruh fitur tiap kelas

3. Mencari nilai *posterior* dari tiap kelas.

Hasil klasifikasi dengan menggunakan metode naïve bayes dilakukan dengan membandingkan nilai *posterior* dari kelas-kelas yang ada. Nilai *posterior* yang paling tinggi yang terpilih sebagai hasil klasifikasi.

Setelah mendapatkan nilai-nilai tersebut, kemudian perlu melihat tingkat akurasi untuk melihat kelayakan sistem. Akurasi merupakan derajat ketepatan antara nilai yang diukur dengan nilai sebenarnya. Nilai replika analis semakin dekat dengan sampel yang sebenarnya maka semakin akurat metode tersebut (Riyanto, 2015) dalam (Argario, Hidayat, & Dewi, 2018). Persamaan akurasi dapat dilihat pada persamaan (5).

$$Akurasi = \frac{\text{Jumlah data benar}}{\text{Total data keseluruhan}} \times 100 \quad (5)$$

### 2.1.1 Langkah Penyelesaian Naïve Bayes

Adapun tahapan dari proses Algoritma Naïve Bayes adalah sebagai berikut (Fauzi, Widodo, & Sunge, 2021).

1. Baca data training
2. Menghitung jumlah kelas per label.
3. Menghitung jumlah kasus per kelas.
4. Kalikan semua variabel kelas.
5. Bandingkan hasil per kelas.

### 2.1.2 Kelebihan Naïve Bayes

Kelebihan Naïve Bayes/ Teori Bayesian adalah sebagai berikut (Fauzi, Widodo, & Sunge, 2021).

1. Mudah untuk dipahami.
2. Hanya memerlukan pengkodean yang sederhana.
3. Lebih cepat dalam perhitungan.
4. Cepat dan efisiensi ruang.

## 2.2 Istilah Dalam Kedokteran

### 2.2.1 Biopsi

*Biopsi* merupakan istilah yang berasal dari bahasa Yunani yaitu “*bios*” yang berarti kehidupan, dan “*opsis*” yang berarti melihat, jadi *biopsi* dapat diartikan sebagai proses melihat kehidupan dari suatu jaringan atau sel yang diambil melalui suatu tindakan medis. *Biopsi* didefinisikan sebagai prosedur medis yang melibatkan pengambilan dan pemeriksaan *mikroskopis* dari jaringan yang berasal dari tubuh manusia yang masih hidup, dengan tujuan untuk menegakkan suatu diagnosis (Erawati, *et al.*, 2017).

### **2.2.2 Histologi**

*Histologi* adalah ilmu yang mempelajari struktur *mikroskopis* sel dan jaringan tubuh yang normal. Ilmu ini merupakan dasar pemahaman *patologi* penyakit, diagnosis dan praktik medis serta penelitian di bidang kedokteran dan kesehatan (Susilowati, Fachiroh, & Sumiwi, 2016). Sedangkan *histopatologi* adalah studi tentang perubahan jaringan akibat penyakit atau gangguan. *Histopatologi* merupakan alat penting dalam diagnosa dan pemahaman penyakit yang mendasari gejala dan kondisi klinis.

## **2.3 Penyakit Kulit Kelompok *Erythemat-Squamous***

Penyakit *erythemat-squamous* merupakan istilah yang digunakan untuk menggambarkan kelompok penyakit yang memiliki ciri khas tertentu. Penyakit kelompok *erythemat-squamous* antara lain.

### **2.3.1 Psoriasis**

*Psoriasis* merupakan penyakit peradangan kulit yang *kronik* dan *residif*, dipengaruhi oleh faktor genetik, dengan karakteristik gangguan pertumbuhan dan diferensiasi epidermis. Penyakit ini dapat muncul pada segala usia, terutama di usia 15-30 tahun. Hingga saat ini pengobatan hanya meredakan gejala sementara (*remisi*), sehingga *psoriasis* sering dianggap sebagai penyakit seumur hidup. Walaupun tidak mengancam nyawa, *psoriasis* dapat mempengaruhi pekerjaan, kehidupan pribadi, dan kualitas hidup pasien. Jika tidak diobati dengan baik, penyakit ini dapat menyebabkan komplikasi dan *komorbiditas*. Gejala biasanya berupa bercak merah bersisik pada kulit, terutama di daerah *ekstensor* dan kulit kepala, disertai rasa gatal. Terkadang pasien juga mengalami nyeri sendi, bercak merah disertai dengan nanah, dan bercak merah bersisik seluruh tubuh. *Psoriasis* dapat dipicu atau memburuk oleh infeksi, obat-obatan, stress, dan merokok. Sering disertai sindrom metabolik. Bisa ditemukan riwayat fenomena *koebner* (Widaty, *et al.*, 2017).

### 2.3.2 *Seborrheic Dermatitis*

*Dermatitis seboroik* adalah suatu kondisi kelainan kulit *papuloskuamosa* yang bersifat kronis dan sering ditemukan pada anak-anak dan orang dewasa. Kondisi ini umumnya terjadi pada area kulit yang memiliki banyak kelenjar sebasea seperti wajah, kulit kepala, telinga, tubuh bagian atas, *fleksura* (*inguinal*, *inframammae*, dan *aksila*) (Widaty, *et al.*, 2017).

### 2.3.3 *Lichen Planus*

*Liken Planus* merupakan penyakit peradangan kronis yang dapat mengenai kulit, kuku, folikel rambut, dan membran mukosa, disertai rasa gatal. Penyakit ini mempunyai karakteristik enam “p” yaitu, *purple* (lesi berwarna keunguan), *polygonal* (poligonal), *pruritic* (gatal), *planar* (permukaan datar), *papules* (papul), dan *plaques* (plak). Lesi *likhen planus* kulit ini berupa lesi yang bilateral atau sering pula bentuknya simetris. *Liken planus* kulit sering terjadi pada usia dewasa muda dan dewasa tua (Gorouhi, Davari, & Fazel, 2014).

### 2.3.4 *Pityriasis Rosea*

*Pityriasis rosea* merupakan suatu kelainan kulit akut yang dimulai dengan munculnya *macula/plak soliter* berwarna merah muda dengan skuama halus (“*herald patch*”), setelah itu dalam beberapa hari hingga beberapa minggu muncul lesi serupa dengan ukuran lebih kecil pada tubuh dan ekstremitas proksimal yang tersusun sesuai lipatan kulit (*christmas tree pattern*) (Widaty, *et al.*, 2017).

### 2.3.5 *Pityriasis Rubra Pilaris*

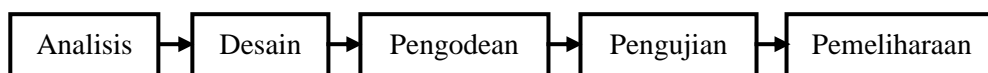
*Pityriasis rubra pilaris (PRP)* merupakan sebuah kondisi kelainan kulit *papuloskuamosa* yang hingga saat ini penyebabnya masih belum diketahui (*idiopatik*). Diagnosis *PRP* dapat ditegakkan berdasarkan pemeriksaan gejala klinis dan *histopatologi*. Meskipun gambaran klinis dapat digunakan untuk menegakkan diagnosis *PRP*, namun karena gambaran klinisnya bervariasi tergantung dari perjalanan penyakit, seringkali pada fase awal *PRP* sulit

dibedakan dengan *psoriasis* yang merupakan diagnosis banding utama (Sari, SP, & Soemarno, 2014).

## 2.4 Model Waterfall

Model SDLC air terjun, yang juga dikenal sebagai model sekuensial linier atau alur hidup klasik, adalah pendekatan perangkat lunak yang berjalan secara berurutan atau sekuensial. Model ini melibatkan tahapan analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*) yang dijalankan secara berurutan.

Berikut merupakan ilustrasi yang menggambarkan model air terjun:



**Gambar 2.1** Model *Waterfall*

- Analisis kebutuhan perangkat lunak  
Proses analisis kebutuhan perangkat lunak melibatkan pengumpulan informasi secara intensif dengan tujuan untuk menentukan secara detail kebutuhan perangkat lunak yang dibutuhkan oleh pengguna. Selama tahap ini, spesifikasi kebutuhan perangkat lunak didokumentasikan untuk keperluan selanjutnya.
- Desain  
Proses desain perangkat lunak melibatkan serangkaian langkah yang bertujuan untuk merancang program perangkat lunak, termasuk struktur data, arsitektur perangkat lunak, antarmuka pengguna, dan prosedur pengodean. Tahap ini menghubungkan kebutuhan perangkat lunak dari tahap analisis kebutuhan menjadi representasi desain yang dapat diimplementasikan dalam bentuk program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan untuk keperluan selanjutnya.
- Pembuatan Kode Program  
Desain perlu diimplementasikan menjadi program perangkat lunak yang sesuai. Pada tahap ini, desain yang telah dibuat akan diubah menjadi kode

program komputer yang dapat dijalankan. Hasil dari tahap ini adalah program perangkat lunak yang sesuai dengan desain yang telah dibuat sebelumnya.

- **Pengujian**

Pengujian bertujuan untuk mengevaluasi perangkat lunak secara menyeluruh, baik dari aspek logika maupun fungsional, dengan tujuan memastikan bahwa semua komponen telah diuji dengan baik. Proses ini dilakukan untuk mengidentifikasi dan mengurangi kesalahan (error) serta memverifikasi bahwa keluaran yang dihasilkan sesuai dengan harapan dan persyaratan yang telah ditetapkan.

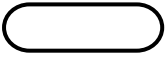

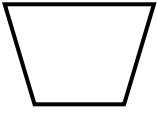
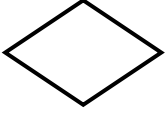

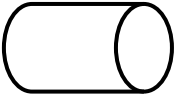
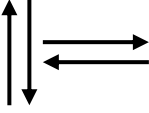
- **Pendukung (*support*) atau pemeliharaan (*maintenance*)**

Pada tahap pendukung atau pemeliharaan, perangkat lunak yang telah dikirimkan kepada pengguna tidak menutup kemungkinan mengalami perubahan. Perubahan tersebut dapat terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi selama pengujian, atau karena perangkat lunak harus beradaptasi dengan lingkungan yang baru. Pada tahap ini, dilakukan analisis spesifik untuk mengidentifikasi perubahan yang perlu dilakukan pada perangkat lunak yang sudah ada, namun tidak digunakan untuk membuat perangkat lunak baru dari awal (S & Shalahuddin, 2013).

## **2.5 Flowchart**

*Flowchart* merupakan bagan yang menunjukkan urutan atau alur dalam suatu program atau prosedur sistem secara logis. *Flowchart* adalah sebuah ilustrasi berupa diagram alir dari algoritma-algoritma dalam suatu program, yang menunjukkan arah aliran dari program tersebut (Yulianeu & Oktamala, 2022).

**Tabel 2.1** Simbol Flowchart




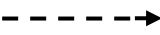
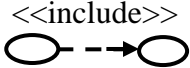
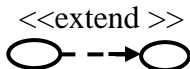
<b>Simbol</b>	<b>Nama</b>	<b>Fungsi</b>
	<i>Terminal</i>	Digunakan untuk memulai atau mengakhiri program.
	<i>Input / Output</i>	Digunakan untuk menyatakan input atau output tanpa melihat jenisnya.
	<i>Manual Operation</i>	Digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	<i>Decision</i>	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu.
	<i>Processing</i>	Digunakan untuk menunjukkan pengolahan data yang dilakukan oleh komputer.
	<i>Disk Storage</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari disk.
	<i>Flow Direction Symbol / Connecting line</i>	Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses.

## 2.6 Use Case Diagram

Menurut Rosa dan Shalahuddin, (2014) dalam (Pitrawati & Sanjaya, 2021) mengemukakan: *Use Case Diagram* merupakan pemodelan untuk melakukan (*behavior*) sistem informai yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut merupakan simbol-simbol yang ada pada diagram *use case*.



**Tabel 2.2** Simbol Use Case

<b>Simbol</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Actor</i>	Pengguna sistem atau yang berinteraksi langsung dengan sistem.
	<i>Use Case</i>	Lingkaran elips dengan nama <i>use case</i> nya tertulis ditengah lingkaran.
	<i>Association</i>	Garis yang berfungsi menghubungkan <i>actor</i> dengan <i>use case</i> .
	<i>Relasi</i>	Sebagai penghubung antara <i>actor usecase</i> , <i>use case-usecase</i> dll.
	<i>Include Relationship</i>	Memungkinkansuatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.
	<i>Extend Relationship</i>	Memungkinkan <i>usecase</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>usecase</i> yang lainnya.

## 2.7 Basis Data

Basis data adalah kumpulan dari item data yang saling berhubungan satu dengan lainnya yang diorganisasikan berdasar sebuah skema atau struktur tertentu. Sistem basis data adalah suatu sistem yang terdiri dari kumpulan tabel data yang saling terhubung dan kumpulan program yang memungkinkan beberapa pengguna atau program lain untuk mengakses dan mengolah tabel-tabel data tersebut (Fathansyah, 1999) dalam (Fikry, 2019). Pengolahan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah pengkat lunak yang disebut DBMS (*Database Management System*) yang menentukan bagaimana data disimpan, diubah dan diambil Kembali (Fikry, 2019).

## 2.8 Dataset

Dataset merupakan sebuah basis data yang berada didalam memori (*in-memory*). Dataset memiliki semua karakteristik, fitur dan fungsi yang ada pada basis data biasa. Dalam dataset dapat ditemukan banyak tabel, dan tabel-tabel dapat memiliki hubungan (*relationship*). Tabel-tabel yang ada di dalam dataset dapat memiliki *foreign key* dan integritas referensial. Dataset berfungsi sebagai objek yang merepresentasikan data dan hubungannya di dalam memori. Strukturnya mirip dengan data yang terdapat dalam basis data (database). Dalam dataset, terdapat koleksi dari tabel-tabel dan data (Yahya & Mahpuz, 2019).

## 2.9 Java

Java adalah sebuah bahasa pemrograman yang pada dasarnya berbasis *object oriented programming* (OOP) atau dalam terjemahan bahasa indonesia-nya lebih dikenal dengan sebutan pemrograman berorientasi objek (PBO). Bahasa ini awalnya dikembangkan oleh James Gosling saat masih bergabung di Sun Microsystems, dan pada tahun 2010, kepemilikan atas java resmi diambil alih oleh oracle (Ahmadian, Mizuardy, & AR, 2017).

## 2.10 Netbeans

Netbeans merupakan alat *Integrated Development Environment* (IDE) yang awalnya dirancang untuk pengembangan perangkat lunak berbasis java, namun juga dapat digunakan untuk bahasa pemrograman lainnya seperti C++ dan PHP (Ahmadian, Mizuardy, & AR, 2017).

## 2.11 Microsoft Excel

Microsoft Excel merupakan perangkat lunak yang dipakai untuk mengolah data secara otomatis yang meliputi penghitungan dasar, penggunaan fungsi dan formula, pembuatan grafik, serta manajemen data. Microsoft Excel termasuk salah satu paket program Microsoft Office yang dikhususkan untuk pengolahan lembar kerja elektronik atau biasa dikenal dengan istilah *spreadsheet* program. *Spreadsheet*

itu sendiri memiliki arti suatu tabel yang nilai-nilainya disusun dalam baris dan kolom (Putri, 2015).

Pada microsoft excel terdapat penyimpanan dalam file dalam format .CSV. format file CSV merupakan kependekan dari *Comma Separated Values*. File CSV adalah format basis data yang dimana setiap perekaman datanya dipisahkan dengan koma (,) maupun titik koma (;). File CSV bisa dibuka menggunakan beberapa aplikasi, seperti notepad, Microsoft Office Exel, dll.

## 2.12 Penelitian Terkait

Berikut ini merupakan beberapa artikel atau jurnal yang terkait dengan penelitian tersebut:

**Tabel 2.3** Penelitian Terkait

No.	Penulis	Judul	Tahun	Keterangan
1.	Yuliyana, Anita Sindar Ros Maryana Sinaga	Sistem Pakar Diagnosa Penyakit Gigi Menggunakan Metode Naive Bayes.	2019	Pada penelitian ini digunakan 37 gejala sebagai masukan dan 7 jenis penyakit sebagai hasil. Hasil klasifikasi metode Bayes menunjukkan bahwa penyakit Halitosis memiliki probabilitas tertinggi, yaitu 0.29646 atau 29.64%
2.	Minarni, Purna Irawan	Implementasi Metode Naive Bayes Untuk Diagnosa Penyakit Lambung.	2019	Penelitian ini menggunakan 5 jenis penyakit lambung dan 18 inputan gejala untuk mendiagnosa penyakit lambung. Pasien dapat melakukan konsultasi dengan menjawab beberapa gejala saja. Namun, Hasil konsultasi masih perlu ditingkatkan akurasi.

				Semakin banyak pengguna yang menjawab pertanyaan pada aplikasi ini, maka tingkat akurasi hasil diagnosa akan meningkat.
3.	Mohammad Syarief, Amirul Mukminin, Novi Prastiti, Wahyudi Setiawan	Penerapan Metode Naïve Bayes Classifier Untuk Deteksi Penyakit Pada Tanaman Jagung.	2017	Hasil uji coba menunjukkan kecocokan deteksi 18/30 dan 17/30 kasus. Penyakit yang paling banyak ditemukan adalah Bulai dan Hama Ulat Grayak. Metode Naïve Bayes kurang efektif, perlu perbaikan dengan menambahkan metode lain seperti Certainty Factor.
4.	Nurmaleni, Ayu Puji Rahayu, Nina Fitriyati	Klasifikasi Jenis Penyakit Erythematous Squamous Berdasarkan Ciri Klinis Dan Histopatologis Menggunakan Metode Analisis Diskriminan Vertex.	2018	Berdasarkan hasil klasifikasi, metode VDA dengan penalti Ridge memiliki nilai APER terkecil dibandingkan penalti lainnya, sehingga lebih tepat digunakan. VDA dengan penalti Euclidian dan Lasso juga berhasil mengklasifikasikan dengan baik. Selanjutnya, dapat dipertimbangkan penggunaan metode VDA non-linier berbasis RKHS untuk membandingkan hasil klasifikasi.
5.	Mhd. Furqan,	Klasifikasi Penyakit Kulit Menggunakan	2022	Berdasarkan hasil penelitian bahwa klasifikasi penyakit kulit (eksim, jerawat, cacar air,

	Yusuf Ramadhan Nasution, Rini Fadillah	Algoritma Naïve Bayes Berdasarkan Tekstur Warna Berdasarkan Android.		campak, dan psoriasis), dapat diidentifikasi melalui metode naïve bayes dapat memperoleh akurasi sebesar 75%.
6.	Mia Martha Amalia, Ernawati, Andang Wijanarko	Implementasi Metode Naïve Bayes Dalam Sistem Pakar Diagnosis Hama dan Penyakit Pada Tanaman Hias Aglaonema SP.	2022	Berdasarkan hasil dan pembahasan, dapat disimpulkan bahwa penelitian ini telah menghasilkan sistem pakar diagnosis hama dan penyakit tanaman hias Aglaonema sp. berbasis web dengan metode Naïve Bayes, yang memiliki akurasi 90% berdasarkan pengujian 30 data.
7.	Siti Khotimatul wildah, Sarifah Agustiani, M. Rangga Ramadhan S, Windu Gata, Hendri Mahmud Nawawi	Deteksi Penyakit Alzheimer Menggunakan Algoritma Naïve Bayes dan Correlation Based Feature Selection.	2020	Algoritma Naïve Bayes dengan seleksi atribut Correlation Based Feature Selection menghasilkan akurasi 94,64% dan AUC 0,945%, sedikit lebih baik dibandingkan Naïve Bayes tanpa seleksi atribut.

8.	Kusnul Kotimah, Albert Yakobus Chandra	Sistem Pakar Penentuan Jenis Kulit Wajah Menggunakan Metode Dempster Shafer.	2022	Hasil dari penelitian ialah, Metode Dempster-Shafer dapat menentukan jenis kulit wajah. Dan hasil Pengujian 50 data customer Omah Milla mencapai akurasi 100%.
9.	Yohanes Bowo Widodo, Silvia Ayu Anggraeini, Tata Sutabri	Perancangan Sistem Pakar Diagnosis Penyakit Diabetes Berbasis Web Menggunakan Algoritma Naive Bayes.	2021	Sistem pakar diagnosis penyakit diabetes menggunakan metode Naive Bayes telah teruji akurat berdasarkan pengujian pada 30 dataset pasien. Penelitian ini masih memiliki kekurangan yang dapat disempurnakan pada penelitian selanjutnya agar aplikasi dapat menjadi lebih sempurna.
10.	Indra Gunaawan, Yusra Fernando	Sistem Pakar Diagnosa Penyakit Kulit Pada Kucing Menggunakan Metode Naive Bayes Berbasis Web.	2021	Aplikasi Sistem Pakar penyakit Kulit Kucing dirancang menggunakan PHP dengan data entry hasil perhitungan Naive Bayes. Pengujian sistem menunjukkan akurasi 80% dibandingkan pakar, sehingga Naive Bayes dapat diterapkan untuk diagnosis penyakit kulit kucing.