

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Harianto, dkk (2018) mengenai Penerapan Algoritma Naive Bayes Untuk Klasifikasi Retensi Arsip, disimpulkan bahwa tingkat keakuratan hasil perhitungan sistem menggunakan algoritma *naive bayes* dengan 1008 data *training* dari pada pengujian yang dilakukan memiliki hasil keakuratan mencapai 95% bila dibandingkan dengan hasil pakar.

Berdasarkan penelitian yang dilakukan oleh Damayanti dan Jenery (2018) mengenai Aplikasi Pengarsipan Surat Dan Data Kelurahan Menggunakan Metode Naïve Bayes Classifier Berbasis E-Document, disimpulkan bahwa hasil akurasi maksimal untuk klasifikasi dokumen didapatkan 85 % dari 200 dokumen sebagai data training. Sehingga semakin tinggi data training semakin tinggi tingkat akurasi.

Berdasarkan penelitian yang dilakukan oleh Rachma dan Wijaya (2019) mengenai Rekomendasi Disposisi Surat dengan Metode *Naïve Bayes* Pada Arsip Surat di Kantor Bakorwil Kabupaten Pamekasan, disimpulkan bahwa Metode *Naïve Bayes Classifier* dapat digunakan untuk melakukanb klasifikasi disposisi perihal surat masuk yang berada di Kantor Bakorwil Pamekasan.

Berdasarkan penelitian yang dilakukan oleh Rachma dan Wijaya (2019) mengenai Rekomendasi Disposisi Surat dengan Metode *Naïve Bayes* Pada Arsip Surat di Kantor Bakorwil Kabupaten Pamekasan, disimpulkan bahwa Metode *Naïve Bayes Classifier* dapat digunakan untuk melakukan klasifikasi disposisi perihal surat masuk yang berada di Kantor Bakorwil Pamekasan.

Berdasarkan penelitian yang dilakukan oleh Trio dan bowo (2019) mengenai Klasifikasi Surat Menggunakan Metode *Naïve Bayes* Pada Sistem Informasi Manajemen Surat, disimpulkan bahwa hasil pengujian sistem didapatkan tingkat akurasi sebesar 83% dengan 6 data latih dan 6 data yang diujikan. Sehingga dari hasil tersebut dapat disimpulkan bahwa input data dengan cara semi manual dapat menghasilkan tingkat akurasi yang cukup tinggi dan mengurangi kesalahan pada data teks.

II.2. Landasan Teori

Landasan teori peneliti kutip dari beberapa teori yang berkaitan dengan penelitian ini yaitu :

II.2.1. Penerapan

Penerapan merupakan proses penterjemahan dari pemodelan menjadi bentuk aplikasi sesuai dengan kebutuhan informasi pengguna. Proses implementasi adalah proses penterjemahan dari pemodelan ke dalam pengkodean atau pembentukan antarmuka. (Fitriani dan Sundari, 2019 : 182).

Penerapan adalah suatu proses ide, konsep, kebijakan atau inovasi dalam suatu tindakan praktis sehingga memberikan dampak baik berupa perubahan pengetahuan, keterampilan maupun nilai dan sikap. Pengertian implementasi secara sederhana implementasi diartikan sebagai pelaksanaan atau penerapan. Implementasi merupakan aktivitas yang saling menyesuaikan. Implementasi merupakan sistem rekayasa. Pengertian-pengertian tersebut memperlihatkan bahwa kata implementasi bermuara pada aktivitas, adanya aksi, tindakan atau mekanisme suatu sistem. Ungkapan mekanisme mengandung arti bahwa implementasi bukan sekedar aktivitas tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh-sungguh berdasarkan acuan norma tertentu untuk mencapai tujuan kegiatan. (Magdalena, dkk, 2021 : 120).

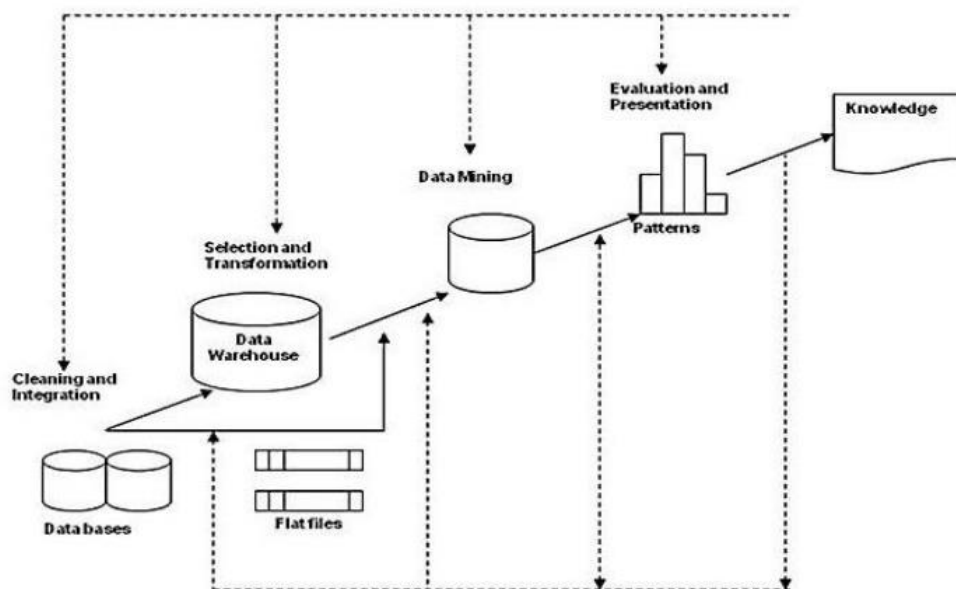
II.2.2. *Data Mining*

Data Mining merupakan proses penggalian data yang sudah ada untuk mendapatkan informasi yang akan diolah kembali menjadi informasi yang baru. (Lianda dan Atmaja, 2021 : 29).

Definisi sederhana dari data mining adalah ekstraksi informasi atau pola yang penting atau menarik dari data yang ada di database yang besar. Dalam jurnal ilmiah, data mining juga dikenal dengan nama *Knowledge Discovery in Database* (KDD). Adapun tahapan *data mining* dapat dilihat pada Gambar II.1. Penjelasan dari setiap tahapannya adalah sebagai berikut :

1. *Data cleaning* (untuk menghilangkan *noise* data yang tidak konsisten).

2. *Data integration* (sumber data yang terpecah dapat disatukan).
 3. *Data selection* (data yang relevan dengan tugas analisis dikembalikan ke dalam database).
 4. *Data transformation* (data berubah atau bersatu menjadi bentuk yang tepat untuk menambang dengan ringkasan performa atau operasi agresi).
 5. *Data mining* (proses esensial dimana metode yang intelejen digunakan untuk mengekstrak pola data).
 6. *Pattern evolution* (untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik).
 7. *Knowledge presentation* (dimana gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambah kepada *user*).
- (Rerung, 2018 : 91).



Gambar II.1. Tahapan Data Mining
(Sumber : Rerung, 2018 : 91)

II.2.3. *Naïve Bayes*

Algoritma *Naïve Bayes Classification* merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat. Proses klasifikasi metode NBC dibagi menjadi dua tahap, yaitu tahap pelatihan dan tahap klasifikasi. Tahap pelatihan dilakukan proses analisis data untuk menghitung jumlah kemunculan kata terhadap sampel data yang sudah diketahui kategorinya. Data-data ini digunakan untuk bahan pembelajaran pada tahap proses klasifikasi untuk menentukan data uji termasuk dalam kategori mana. (Trisetiyo, dkk, 2019 : 286). Rumus metode naive bayes dapat dilihat pada persamaan

II.1

$P(C|X) =$

$$\frac{P(X|C_i) * P(C_i)}{P(X)} \dots\dots\dots (II.1)$$

Dimana :

$P(C|X)$: Probabilitas Kejadian C dan X

$P(X|C)$: Probabilitas Kejadian X terhadap C_i

$P(C_i)$: Probabilitas Kejadian C_i

$P(X)$: Probabilitas Kejadian X. (Lianda dan Atmaja, 2021 : 30).

II.2.4. Pengarsipan

Arsip merupakan kumpulan warkat yang disimpan secara sistematis karena mempunyai kegunaan agar setiap kali diperlukan dapat secara cepat ditemukan kembali. Arsip mempunyai peranan yang penting dalam sebuah kantor, apabila arsip yang dimiliki kurang baik pengelolaannya dapat mengakibatkan sulitnya menemukan informasi yang telah disimpan dan akhirnya dapat menghambat tahapan proses pekerjaan selanjutnya. (Bachtiar dan Jemakmun, 2021 : 396).

Arsip terdiri dari 2 jenis, antara lain :

1. Arsip Konvensional; contoh: arsip kertas
2. Arsip Media Baru; contoh: arsip micro film, kaset, dan lain-lain

Arsip berasal dari bahasa Belanda, *archieff*. *archieff* dalam bahasa Belanda memiliki beberapa pengertian berikut ini :

1. Tempat penyimpanan secara teratur bahan-bahan arsip : bahan-bahan tertulis, piagam-piagam, surat-surat, keputusan-keputusan, akte-akte, daftar-daftar, dokumen-dokumen, peta-peta.
2. Kumpulan teratur, dari bahan-bahan kearsipan tersebut.
3. Bahan-bahan yang harus diarsip itu sendiri.

Dalam bahasa inggris, arsip dinyatakan dengan istilah *file*, yang berasal dari bahasa latin *filum* yang berarti tali atau benang. Karena pada awalnya orang-orang inggris menyatakan warkat dengan cara mengikatnya dengan tali atau benang. (Romindo dan Ameylia, 2019 : 82).

II.2.5. *Hyper Text Markup Language (HTML)*

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang user untuk membuat dan menyusun bagian paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk halaman web dan aplikasi. HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan user untuk mengorganisir dan memformat dokumen, sama seperti Microsoft Word. (Sugijanto, dkk, 2020 : 2).

HTML ialah kepanjangan dari *Hypertext Markup Language*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui tag-tag yang ditulis dalam HTML. (Rahmasari, 2019 : 415).

II.2.6. *Hypertext Preprocessor (PHP)*

PHP merupakan singkatan dari “Hypertext Preprocessor”. PHP adalah sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrograman C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa file dalam 1 file dengan cara di include dan require. PHP itu sendiri sudah dapat berinteraksi dengan beberapa database walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle. (Rahmasari, 2019 : 415).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script/program* tersebut akan dijalankan/diproses oleh *server*. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. (Sugijanto, dkk, 2020 : 2).

II.2.7. My Structure Query Language (MySQL)

Definisi MySQL merupakan software RDMS (Relational Database Management System) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak pengguna dan dapat melakukan suatu proses secara sinkron atau bersamaan. (Rahmasari, 2019 : 415).

MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database* pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial

untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.
(Sugijanto, dkk, 2020 : 2).

II.2.8. *Unified Modeling Language (UML)*

UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).

II.2.9. *Penelitian Terkait*

Penelitian yang dikait dengan peelitian yang sudah dilakukan sebelumnya adalah sebagai berikut :

- a. Menurut Purnomo (2013) dalam penelitiannya menyimpulkan bahwa dengan adanya sistem informasi pendaftaran pernikahan ini pencarian data dilakukan dengan lebih mudah, pembuatan laporan juga lebih cepat dan menghasilkan data yang akurat
- b. Menurut perdana (2014) dalam penelitiannya menyimpulkan bahwa sistem informasi adminitrasi pernikahan ini dapat membantu Kantor Uurusan Agama dalam meningkatkan kualitas kinerja pelayanan masyarakat dibagian

pendaftaran dan pencatatan pernikahan yang semula masih memerlukan waktu yang cukup lama dalam proses pendataan dan pencatatan nikah. Setelah adanya sistem ini dapat menjadi solusi bagi Kantor Urusan Agama Kecamatan Medan Delikarena dapat memberikan pelayanan yang cepat dan efisien

c. Menurut Muhasshanah (20217) dalam penelitiannya menyimpulkan bahwa

sistem informasi data pernikahan berbasis web yang dibangun mempermudah pegawai KUA dalam memproses data-data pendaftaran pernikahan secara mudah dan menghasilkan sistem pernikahan yang efektif dan efisien

d. Menurut Givo (2016) dalam penelitiannya menyimpulkan bahwa sistem

informasi pendaftaran pernikahan berbasis *online* yang dibangun dapat mempermudah pelayanan calon pengantin dalam mendaftarkan pernikahan

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan

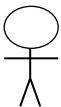
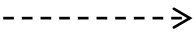
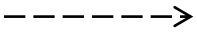
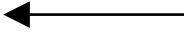
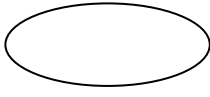
UML adalah sebagai berikut :

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol *Use Case*

Gambar	Nama	Keterangan
--------	------	------------

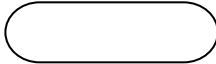
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber : Andikos, 2019 : 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol *Activity Diagram*

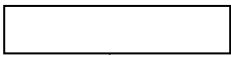
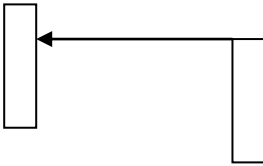
Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber : Andikos, 2019 : 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence Diagram dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

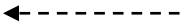
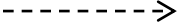
(Sumber : Andikos, 2019 : 39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang

		berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Assocation</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Andikos, 2019 : 39)