

**KLASIFIKASI *MESSAGE SPAM* MENGGUNAKAN  
METODE *NEIVE BAYES* DAN *MULTINOMIAL NAÏVE BAYES***

**TUGAS AKHIR**

**Diajukan Oleh :**

**CUT ELFIA ZAHARA**

**NIM. 190705043**

**Mahasiswa Fakultas Sains dan Teknologi**

**Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI AR-RANIRY  
BANDA ACEH  
2023 M/1445 H**

## LEMBAR PERSETUJUAN

### KLASIFIKASI *MESSAGE* MENGGUNAKAN METODE *NAÏVE BAYES* DAN *MULTINOMIAL NAÏVE BAYES*

#### TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh  
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana  
pada Prodi Teknologi Informasi

Oleh:

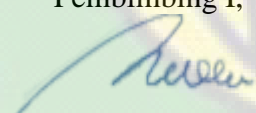
**CUT ELFIA ZAHARA**

**NIM. 190705043**

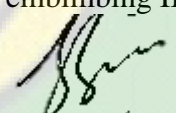
**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**

Disetujui untuk Dimunaqasyahkan Oleh:

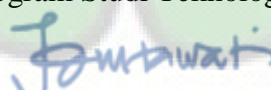
Pembimbing I,

  
**Hendri Ahmadian, M.I.M**  
**NIP. 198301042014031002**

Pembimbing II,

  
**Bustami, M.Sc**  
**NIP. 198604082014031001**

Mengetahui,  
Ketua Program Studi Teknologi Informasi

  
**Ima Dwitawati, MBA**  
**NIP. 198210132014032002**

## LEMBAR PENGESAHAN

### KLASIFIKASI MESSAGE MENGGUNAKAN METODE NAÏVE BAYES DAN MULTINOMIAL NAÏVE BAYES

#### TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir  
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus  
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)  
Pada Prodi Teknologi Informasi

Pada Hari/Tanggal: Selasa, 19 Desember 2023  
06 Jumadil Akhir 1445 H

di Darussalam, Banda Aceh  
Panitia Ujian Munaqasah Tugas Akhir

Ketua,

Hendri Ahmadian M.I.M  
NIP. 198301042014031002

Sekretaris,

Bustami, M.Sc  
NIP. 198604082014031001

Penguji I,

Malahayati, MT  
NIP. 19830127015032003

Penguji II,

Khairan AR, M.Kom  
NIP. 198607042014031001

Mengetahui,  
Dekan Fakultas Sains dan Teknologi  
UIN Ar-Raniry Banda Aceh

Dr. Ir. M. Dirhamsyah, M.T., IPU  
NIP.196210021988111001

## LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Cut Elfia Zahara  
NIM : 190705043  
Program Studi : Teknologi Informasi  
Fakultas : Sains dan Teknologi  
Judul Tugas Akhir : Klasifikasi *Message* Menggunakan Metode *Naïve Bayes*  
dan *Multinomial Naïve Bayes*

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu mempertanggungjawab atas karya ini;

Bila kemudian hari ini ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat mempertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenakan sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 19 Desember 2023

Yang Menyatakan



Cut Elfia Zahara

## ABSTRAK

Nama : Cut Elfia Zahara  
NIM : 190705043  
Program Studi : Teknologi Informasi  
Judul : Klasifikasi *Message* Menggunakan Metode *Naïve Bayes* dan *Multinomial Naïve Bayes*  
Tanggal Sidang : 19 Desember 2023  
Jumlah Halaman : 65 Halaman  
Pembimbing I : Hendri Ahmadian, M.I.M  
Pembimbing II : Bustami, M.Sc

Perkembangan teknologi informasi semakin pesat. Salah satu penggunaan teknologi informasi yaitu sebagai *platform* komunikasi digital yang telah menciptakan tantangan baru terkait pesan *spam* yang dapat merugikan pengguna. Penelitian ini memfokuskan pada pengembangan metode klasifikasi untuk membedakan pesan *spam* dan *non-spam* menggunakan metode *Naïve Bayes* dan *Multinomial Naïve Bayes*.

Dalam penelitian ini mengumpulkan dataset dari berbagai sumber komunikasi seperti *WhatsApp*, *Email*, dan SMS. Implementasi metode *Naïve Bayes* menunjukkan tingkat akurasi yang signifikan, mencapai 89% untuk dataset *WhatsApp*, 99% untuk dataset *Email*, dan 98% untuk dataset SMS. Sementara itu, metode *Multinomial Naïve Bayes* memberikan akurasi sebesar 81% untuk dataset *WhatsApp*, 87% untuk dataset *Email*, dan 96% untuk dataset SMS. Hasil penelitian ini menunjukkan bahwa metode klasifikasi yang diusulkan dapat meminimalisir penyebaran pesan *spam*, melindungi pengguna dari ancaman *malware*, serta membantu pengelompokan pesan dengan baik sesuai dengan keasliannya. Penelitian ini memberikan kontribusi penting dalam meningkatkan keamanan dan kenyamanan pengguna dalam berkomunikasi melalui *platform* digital.

**Kata Kunci:** Klasifikasi, *Naïve Bayes*, *Multinomial Naïve Bayes*, Akurasi.

## ABSTRACT

Name : Cut Elfia Zahara  
Student Number : 190705043  
Department : Information Technology  
Title : Message Classification Using Naïve Bayes and  
Multinomial Naïve Bayes Methods  
Date : 19 December 2023  
Number of Pages : 65 Pages  
Supervisor I : Hendri Ahmadian, M.I.M  
Supervisor II : Bustami, M.Sc

The rapid development of information technology has led to the emergence of new challenges in the form of spam messages, which can be harmful to users. This research focuses on developing classification methods to distinguish between spam and non-spam messages using Naïve Bayes and Multinomial Naïve Bayes techniques.

In this study, datasets were collected from various communication sources such as WhatsApp, Email, and SMS. The implementation of the Naïve Bayes method demonstrated significant accuracy levels, reaching 89% for the WhatsApp dataset, 99% for the Email dataset, and 98% for the SMS dataset. Meanwhile, the Multinomial Naïve Bayes method achieved accuracies of 81% for the WhatsApp dataset, 87% for the Email dataset, and 96% for the SMS dataset. The research findings indicate that the proposed classification methods can minimize the spread of spam messages, protect users from malware threats, and effectively categorize messages based on their authenticity. This study provides a valuable contribution to enhancing the security and user experience in digital communication platforms.

**Keywords:** Classification, Naïve Bayes, Multinomial Naïve Bayes, Accuracy.

## KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penyayang. Puji syukur penulis panjatkan panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “**Klasifikasi Message Menggunakan Metode Naïve Bayes dan Multinomial Naïve Bayes**”. Shalawat serta salam kepada Rasulullah SAW yang mengantarkan manusia dari zaman kegelapan ke zaman yang terang benderang.

Dalam proses penyusunan tugas akhir ini penulis menyadari bahwa tidak dapat menyelesaikan tugas akhir jika tidak adanya dukungan, bimbingan, motivasi dan bantuan dari berbagai pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini. Dengan kerendahan hati penulis ingin berterimakasih kepada :

1. Kedua orangtua yang senantiasa memberikan dukungan dan doa kepada saya dalam menyusun tugas akhir ini.
2. Ketua dan Sekretaris Prodi Teknologi Informasi Ibu Ima Dwitawati, M.B.A dan Bapak Khairan AR, M.Kom yang telah memberikan bimbingan dan arahan dalam penyusunan tugas akhir ini.
3. Bapak Hendri Ahmadian, M.I.M selaku Pembimbing I dan Bapak Bustami, M.Sc selaku Pembimbing II yang senantiasa memberikan arahan dan bimbingan dalam penyusunan tugas akhir ini.
4. Ibu Cut Ida Rahmadiana, S.Si selaku *Staff* Prodi Teknologi Informasi, yang senantiasa membantu penulis dalam pemberkasan administrasi.
5. Bapak Dr. Ir. M. Dirhamsyah, M.T.,IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry yang telah mendukung dan memberi motivasi untuk kami.
6. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu kepada penulis dalam bidang Teknologi Informasi.
7. Sahabat dan teman-teman yang selalu memberikan dukungan moral dalam menyelesaikan penyusunan tugas akhir ini.

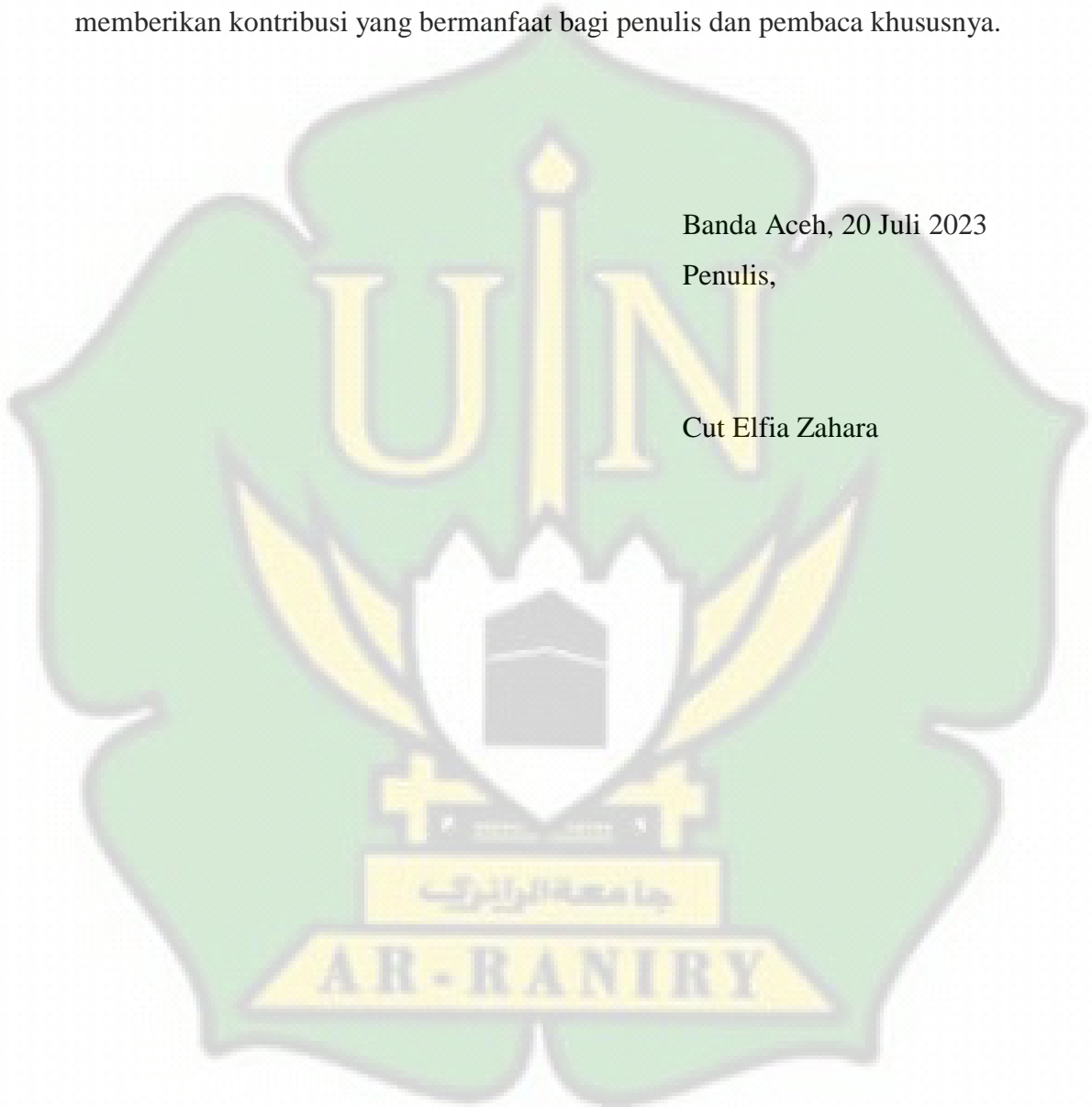
8. Semua pihak yang telah membantu dalam menyelesaikan proposal tugas akhir yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dari tugas akhir ini. Oleh karena itu, penulis memohon maaf atas segala kekurangan yang terjadi selama proses penyusunan tugas akhir. Akhir kata, semoga tugas akhir ini dapat memberikan kontribusi yang bermanfaat bagi penulis dan pembaca khususnya.

Banda Aceh, 20 Juli 2023

Penulis,

Cut Elfia Zahara

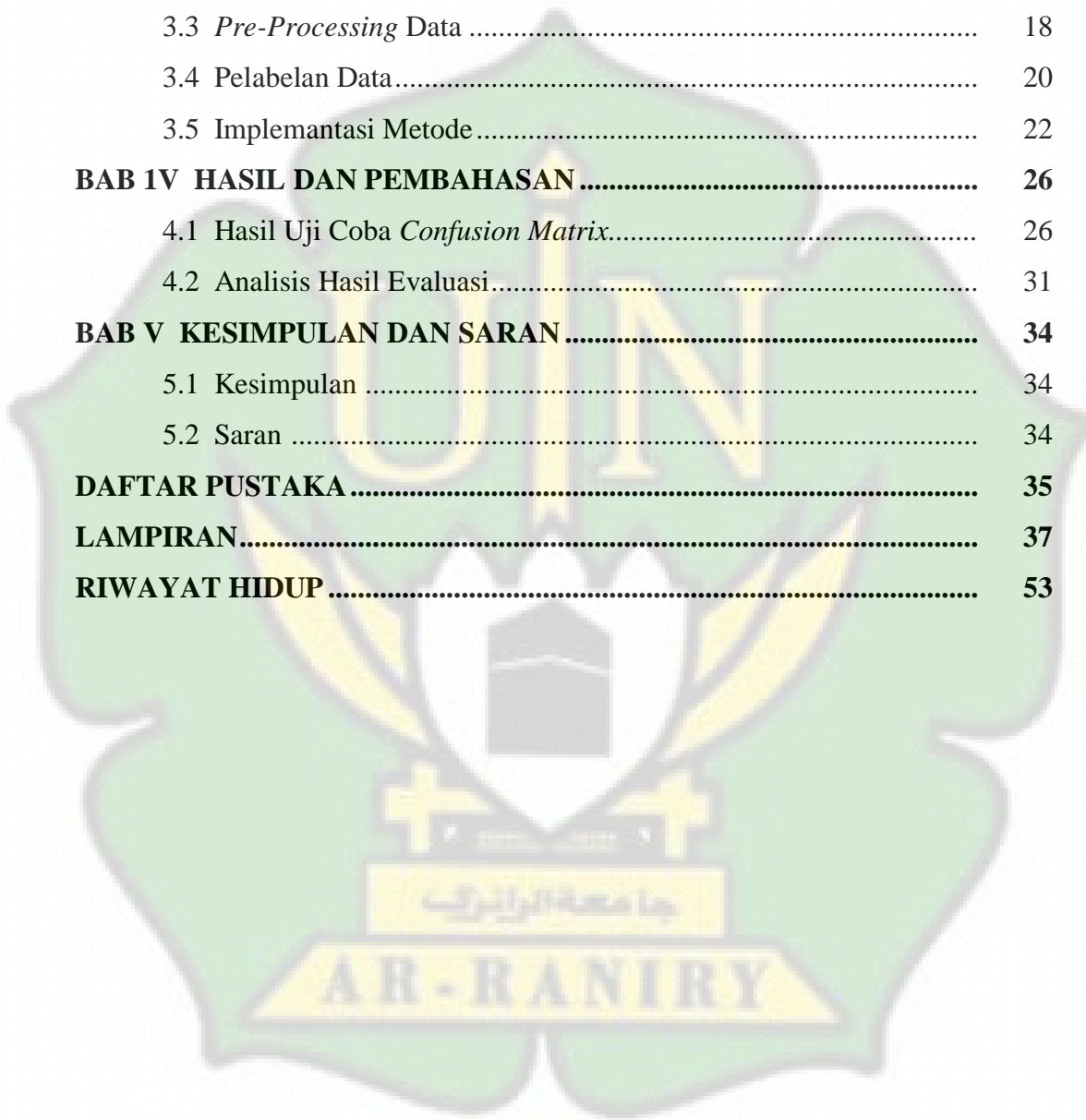




## DAFTAR ISI

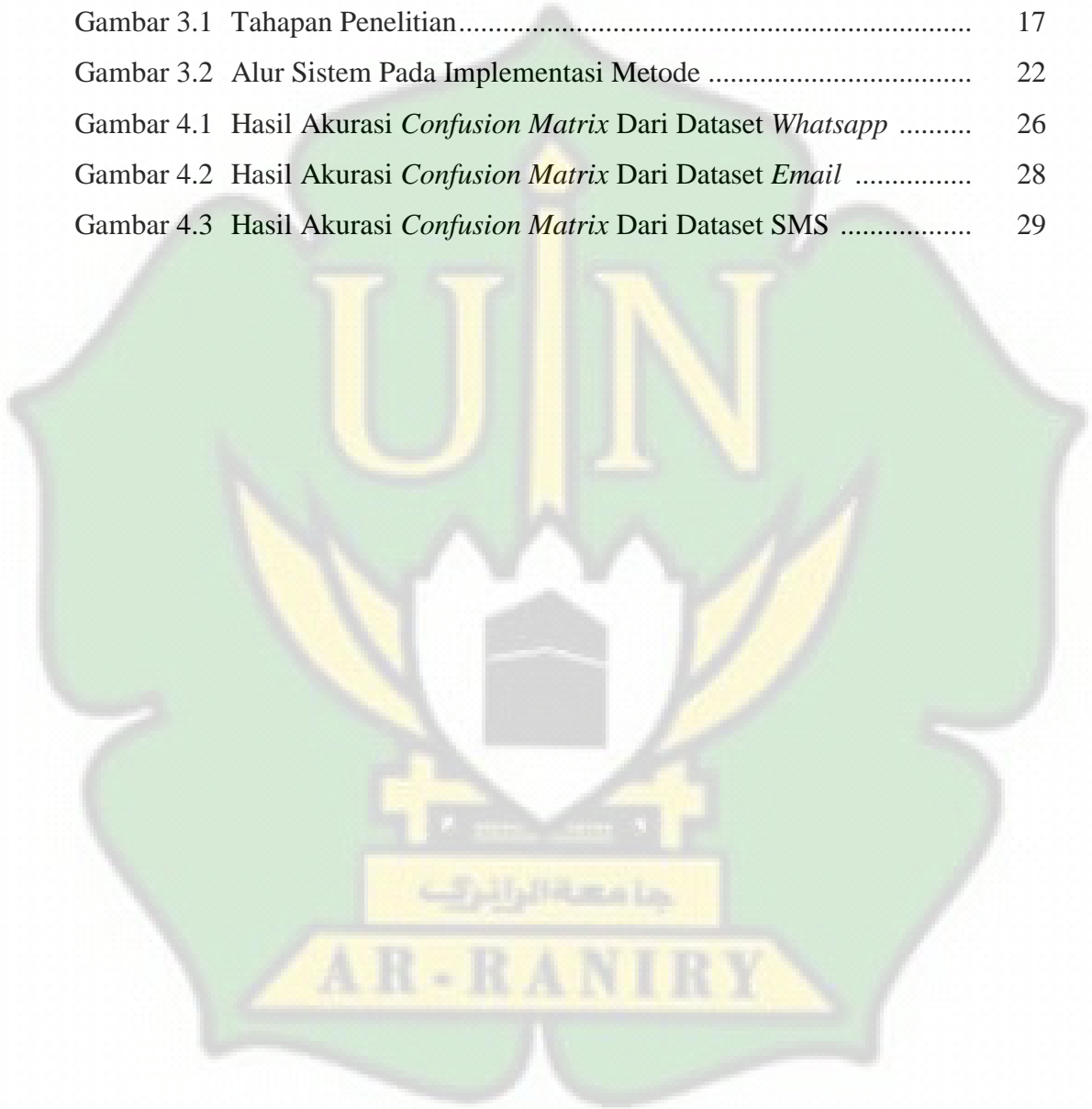
<b>LEMBARAN PERSETUJUAN .....</b>	<b>i</b>
<b>LEMBARAN PENGESAHAN.....</b>	<b>ii</b>
<b>LEMBARAN PERNYATAAN KEASLIAN.....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>iv</b>
<b>ABSTRACT .....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL.....</b>	<b>xi</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>5</b>
2.1 Penelitian Terdahulu .....	5
2.2 Klasifikasi.....	7
2.3 <i>Message</i> (Pesan).....	8
2.3.1 <i>WhatsApp Message</i> .....	8
2.3.2 <i>Short Message Service (SMS)</i> .....	9
2.3.3 <i>Email</i> .....	10
2.4 <i>Machine Learning</i> .....	10
2.5 <i>Natural Language Processing (NLP)</i> .....	11
2.6 <i>TensorFlow</i> .....	11
2.7 <i>Naïve Bayes</i> .....	12
2.8 <i>Multinomial Naïve Bayes</i> .....	13
2.9 <i>Confusion Matrix</i> .....	14
2.10 <i>Tools</i> .....	16

2.10.1 <i>Python</i> .....	16
2.10.2 <i>Google Colaboratory</i> .....	16
<b>BAB III METODE PENELITIAN .....</b>	<b>17</b>
3.1 Tahapan Penelitian .....	17
3.2 Pengumpulan Data .....	18
3.3 <i>Pre-Processing</i> Data .....	18
3.4 Pelabelan Data.....	20
3.5 Implementasi Metode.....	22
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>26</b>
4.1 Hasil Uji Coba <i>Confusion Matrix</i> .....	26
4.2 Analisis Hasil Evaluasi.....	31
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>34</b>
5.1 Kesimpulan .....	34
5.2 Saran .....	34
<b>DAFTAR PUSTAKA .....</b>	<b>35</b>
<b>LAMPIRAN.....</b>	<b>37</b>
<b>RIWAYAT HIDUP .....</b>	<b>53</b>



## DAFTAR GAMBAR

Gambar 2.1 <i>Machine Learning</i> Sebagai Cabang Ilmu <i>Artificial Intelligence</i> (AI).....	11
Gambar 3.1 Tahapan Penelitian.....	17
Gambar 3.2 Alur Sistem Pada Implementasi Metode .....	22
Gambar 4.1 Hasil Akurasi <i>Confusion Matrix</i> Dari Dataset <i>Whatsapp</i> .....	26
Gambar 4.2 Hasil Akurasi <i>Confusion Matrix</i> Dari Dataset <i>Email</i> .....	28
Gambar 4.3 Hasil Akurasi <i>Confusion Matrix</i> Dari Dataset SMS .....	29



## DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Terdahulu .....	6
Tabel 2.2 <i>Confusion Matrix</i> .....	14
Tabel 3.1 Contoh <i>Case Folding</i> .....	18
Tabel 3.2 Contoh Tokenisasi.....	19
Tabel 3.3 Contoh Mengubah Menjadi <i>Lowercase</i> .....	19
Tabel 3.4 Contoh Menghilangkan <i>Stopwords</i> .....	19
Tabel 3.5 Contoh <i>Stemming</i> .....	20
Tabel 3.6 Contoh Pelabelan Kalimat.....	20
Tabel 4.1 Perhitungan <i>Confusion Matrix</i> menggunakan dataset <i>WhatsApp</i> .	27
Tabel 4.2 Hasil akurasi, <i>recall</i> , <i>preciission</i> dan <i>f1-score</i> dari dataset <i>Whatsapp</i> .....	27
Tabel 4.3 Perhitungan <i>Confusion Matrix</i> menggunakan dataset <i>Email</i> .....	28
Tabel 4.4 Hasil akurasi, <i>recall</i> , <i>preciission</i> dan <i>f1-score</i> dari dataset <i>Email</i> .	28
Tabel 4.5 Perhitungan <i>Confusion Matrix</i> menggunakan dataset SMS.....	30
Tabel 4.6 Hasil akurasi, <i>recall</i> , <i>preciission</i> dan <i>f1-score</i> dari dataset SMS...	30
Tabel 4.7 Perbandingan Model <i>Naïve Bayes</i> dan <i>Multinomial Naïve Bayes</i>	31
Tabel 4.8 Grafik Perbandingan Akurasi Metode <i>Naïve Bayes</i> dan <i>Multinomial Naïve Bayes</i> .....	33

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi informasi semakin pesat. Perkembangan teknologi ini memudahkan setiap individu untuk berkomunikasi. Perkembangan media komunikasi saat ini telah memiliki berbagai *platform* diantaranya yaitu *Facebook, Twitter, Line, Instagram, Short Message Service (SMS), Email* dan *WhatsApp*.

Pesan merupakan bagian dari unsur-unsur komunikasi, *Hafied Cangara* dalam bukunya *Pengantar Ilmu Komunikasi* menyatakan bahwa “ Dalam proses komunikasi, pengertian pesan adalah sesuatu yang disampaikan pengirim kepada penerima. Pesan dapat disampaikan dengan cara tatap muka atau melalui media komunikasi. Isinya bisa berupa ilmu pengetahuan, hiburan, informasi, nasihat atau propaganda”. Pengertian pesan itu sendiri menurut *Onong Uchjana Effendy* merupakan terjemahan dari bahasa asing “*message*” yang artinya adalah lambang bermakna (*meaningful symbols*), yaitu lambang yang membawakan pikiran atau perasaan komunikator (*Bahari, 2022*). Pesan teks ini terbagi kedalam beberapa jenis yaitu pesan digital, pesan media sosial dan pesan formal dan informal.

Seringkali pesan yang diterima *user* tidak hanya berupa pesan pribadi yang berasal dari sesama *user*, akan tetapi pesan yang masuk dapat berasal dari perusahaan dengan tujuan memberikan informasi perusahaan tersebut. Namun pesan ini bisa dikategorikan pesan *spam* karena berupa pesan promosi atau iklan. Akibatnya, antara pesan penting dan pesan *spam* tidak terfilterisasi. Pesan *Spam* merupakan pesan yang tidak diinginkan, sering kali dikirim secara massal dan tidak sah. Pesan *spam* dapat berisi tautan berbahaya, penipuan, atau promosi yang tidak diminta.

Salah satu fenomena pesan *spam* yang sekarang sedang marak dibicarakan yaitu pesan *spam* berbentuk undangan dengan format.apk. Adapun salah satu keluhan masyarakat mengenai pesan yang mengandung undangan.apk dapat menjadi sarana penyebaran *malware* atau virus yang dapat merusak perangkat

atau mencuri data pribadi *user*. Pesan undangan.apk dapat digunakan untuk melakukan penipuan dengan tujuan mendapatkan informasi data pribadi dan penipu juga bisa membuat rekening milik korban terkuras. Sehingga diperlukan sebuah metode klasifikasi untuk pesan *spam* dan *non-spam* dalam mengatasi permasalahan gangguan komunikasi. Pesan yang seharusnya berasal dari kontak *WhatsApp* dapat tersembunyi diantara pesan *spam* yang tidak relevan. Salah satu cara yang bisa dilakukan untuk membedakan antara *spam* dan *non-spam* adalah dengan tehnik klasifikasi.

Klasifikasi merupakan pengelompokan suatu objek ke dalam kelas tertentu. Klasifikasi adalah salah satu metode yang ada didalam *data mining* yang digunakan untuk mengelompokan suatu objek kedalam suatu kelas tertentu. Salah satu metode yang bisa diterapkan untuk melakukan proses klasifikasi adalah metode *Naïve Bayes*. *Naïve Bayes* merupakan salah satu metode klasifikasi yang memiliki kelebihan dalam perhitungan dengan algoritma yang relatif sederhana dan menghasilkan nilai akurasi yang tinggi (Putro dkk., 2020).

Penelitian mengenai klasifikasi pesan *spam* itu sudah pernah dilakukan sebelumnya, salah satunya adalah penelitian yang dilakukan oleh (Hidayat, 2023). Pada penelitian ini peneliti menggunakan metode *Naïve Bayes* dalam mengklasifikasi *spam email*. Dataset yang digunakan berasal dari 5 email, yang terdiri dari 2 *email spam* dan 3 *email ham (non-spam)* . Hasil yang diperoleh dari penelitian ini didapat akurasi di dalam klasifikasi pesan *spam* sebanyak 60%. Namun kekurangan dari penelitian ini yaitu jumlah dataset yang digunakan terlalu sedikit sehingga hasil dari penelitian ini masih sangat diragukan. Selain itu, penelitian yang serupa juga dilakukan oleh (Mustofa & Mahfudh, 2019) yaitu penelitian untuk mengklasifikasi berita *hoax* dengan menggunakan metode *Naïve Bayes*. Penelitian ini menggunakan 151 data, hasil yang diperoleh dari penelitian ini yaitu berupa akurasi yang mencapai 85,28%. Adapun kekurangan dari penelitian ini adalah dataset yang masih tergolong terbatas dan perlu penambahan jumlah dataset.

Pada penelitian lain yang dilakukan oleh (Dewi, 2021). Penelitian ini menerapkan metode *Multinomial Naïve Bayes* untuk melakukan klasifikasi berita dari berbagai sumber, termasuk surat kabar, majalah, radio dan televisi. Sampel

data yang digunakan dalam penelitian ini berjumlah 10.500 data, dengan hasil akurasi mencapai 96%. Namun dari hasil penelitian ini peneliti menyarankan untuk menambahkan lagi sampel kategori artikel berita sesuai dengan kriteria kategori yang berkembang saat ini.

Berdasarkan uraian di atas, penulis berinisiatif melakukan penelitian dengan topik “*Klasifikasi Message Spam dan non-Spam Menggunakan Metode Naïve Bayes dan Multinomial Naïve Bayes*”. Dari hasil penelitian ini diharapkan dapat meminimalisir terjadinya penyebaran virus berbahaya dan penipuan lainnya, sehingga pesan *spam* dan *non-spam* dapat dikelompokkan dengan baik.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah penulis uraikan, maka penulis merumuskan masalah yang ada yaitu sebagai berikut:

1. Bagaimana proses klasifikasi *message spam* menggunakan metode *Naïve Bayes* dan *Multinomial Naïve Bayes*?
2. Bagaimana nilai akurasi dari hasil klasifikasi *message spam* dengan metode *Naïve Bayes* dan *Multinomial Naïve Bayes*?

## **1.3 Tujuan Penelitian**

Berdasarkan latar belakang dan rumusan masalah, maka tujuan dari penelitian ini adalah :

1. Mengetahui proses pengimplementasian metode *Naïve Bayes* dan *Multinomial Naïve Bayes* pada klasifikasi *message spam*.
2. Mengetahui nilai akurasi dari hasil klasifikasi *message spam* dengan metode *Naïve Bayes* dan *Multinomial Naïve Bayes*.

## **1.4 Batasan Penelitian**

Dari rumusan masalah yang telah diuraikan, maka diambil beberapa batasan masalah sebagai berikut :

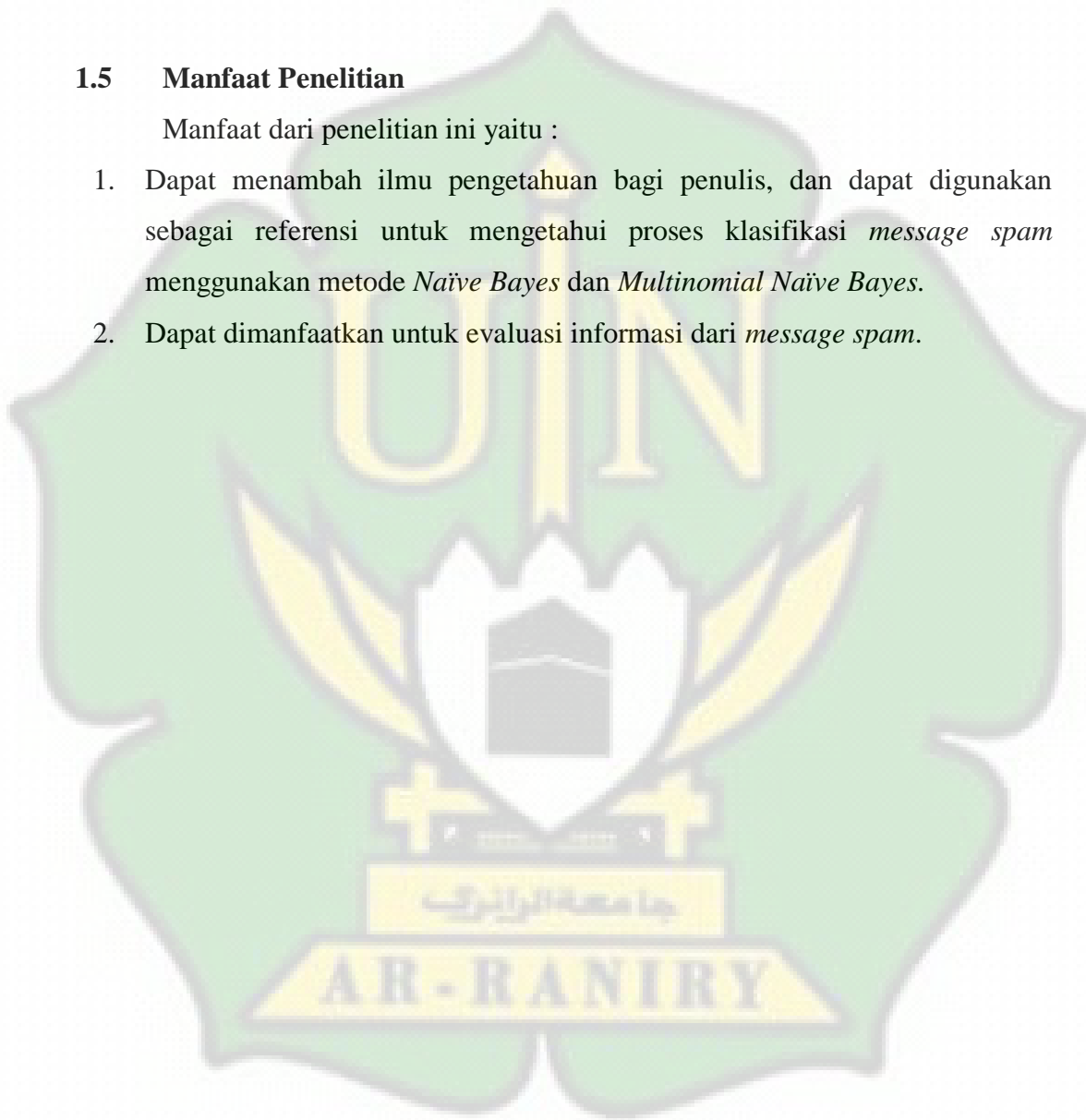
1. Data yang digunakan merupakan kumpulan dari 3 *message* yang berbeda yaitu data *WhatsApp*, *Email* dan *SMS*.

2. Metode klasifikasi yang digunakan pada penelitian ini adalah *Naïve Bayes* dan *Multinomial Naïve Bayes*.
3. *Tools* yang digunakan pada penelitian ini yaitu menggunakan bahasa pemrograman *python* dengan *framework TensorFlow* dan teks editor *Google Colaboratory*.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu :

1. Dapat menambah ilmu pengetahuan bagi penulis, dan dapat digunakan sebagai referensi untuk mengetahui proses klasifikasi *message spam* menggunakan metode *Naïve Bayes* dan *Multinomial Naïve Bayes*.
2. Dapat dimanfaatkan untuk evaluasi informasi dari *message spam*.





## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Berkaitan dengan penelitian yang dilakukan oleh penulis dengan menggunakan metode klasifikasi *Naïve Bayes* dan *Multinomial Naïve Bayes* penelitian terdahulu yang relevan dengan penelitian yang dilakukan penulis sangat dibutuhkan sebagai referensi dalam mengembangkan penelitian yang akan dilakukan oleh penulis.

Dalam penelitian mengenai klasifikasi menggunakan metode *Naïve Bayes*. Pertama, penelitian dengan topik “*Klasifikasi Berita Hoax Dengan Menggunakan metode Naive Bayes*” (Mustofa & Mahfudh, 2019). Hasil dari penelitian ini memperoleh nilai akurasi sebesar 85.28% yang mana dokumen terklasifikasi yang relevan sebanyak 307 dan dokumen yang tidak relevan sebanyak 53 atau *error rate* sebesar 14.72%.

Kedua, penelitian dengan topik “*Penerapan Metode Naive Bayes Untuk Klasifikasi Pelanggan*” yang dilakukan oleh (Putro dkk., 2020). Penelitian ini memperoleh nilai akurasi sebanyak 92% dengan jumlah data sebanyak 100, dimana sebanyak 75 data latih dan 25 data uji. Ketiga, penelitian dengan judul “*Klasifikasi Pesan Gangguan Pelanggan Menggunakan Metode Naive Bayes Classifier*” yang diteliti (Haryono dkk., 2018). Penelitian ini memperoleh nilai akurasi sebanyak 95% dan membuktikan bahwa metode *Naïve Bayes* dapat digunakan untuk membantu klasifikasi pesan dari *user*.

Keempat, penelitian yang dilakukan oleh (Fitria & Azis, 2018) dengan judul “*Analisis Kinerja Sistem Klasifikasi Skripsi menggunakan Metode Naïve Bayes Classifier*”. Hasil dari penelitian memperoleh nilai akurasi sebanyak 50%. Data yang digunakan pada penelitian ini adalah data skripsi yang ada di perpustakaan Fakultas Ilmu Komputer. Kelima, penelitian oleh (Hidayat, 2023) dengan judul “*Klasifikasi Spam Email Menggunakan Metode Naïve Bayes*”. Dari penelitian ini memperoleh nilai akurasi sebanyak 60% dengan mengambil data *training* yang terdiri dari 5 *email spam* dan *ham (non-spam)*.

Keenam, pada penelitian yang dilakukan oleh (Dewi, 2021). Dengan judul “Klasifikasi Berita Menggunakan Metode Multinomial Naïve Bayes”. Penelitian ini memperoleh nilai akurasi sebanyak 96%. Pengujian dilakukan dengan menggunakan 10.500 dataset dan tujuh kategori berita, seperti polhukam, bisnis, olahraga, hiburan, teknologi, otomotif dan kesehatan.

Tabel 2.1 Perbandingan Penelitian Terdahulu

Penelitian	Metode	Judul Penelitian	Dataset	Akurasi
(Mustofa & Mahfudh, 2019)	Naive Bayes	Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes	Dataset yang digunakan sebanyak 300 data terdapat 297 dokumen berita berkategori hoax dan 63 dokumen berkategori fakta.	85.28%
(Putro dkk., 2020)	Naïve Bayes	Penerapan Metode Naive Bayes Untuk Klasifikasi Pelanggan	Dataset yang digunakan sebanyak 100 data, yang terdiri dari data penjualan dari bulan juni - agustus	92%
(Haryono dkk., 2018)	Naive Bayes	Klasifikasi Pesan Gangguan Pelanggan Menggunakan Metode Naive Bayes Classifier	Dataset yang digunakan sebanyak 120, terdiri dari pesan gangguan pelanggan	95%
(Fitria & Azis, 2018)	Naïve Bayes	Analisis Kinerja Sistem Klasifikasi Skripsi menggunakan Metode Naive	Dataset yang digunakan pada penelitian ini dari 15 judul skripsi.	50%

Penelitian	Metode	Judul Penelitian	Dataset	Akurasi
		Bayes Classifier		
(Hidayat, 2023)	Naïve Bayes	Klasifikasi Spam Email Menggunakan Metode Naïve Bayes	Dataset pada penelitian ini yaitu dari 5 email yang terdiri dari 2 email spam dan 3 email ham.	60%
(Dewi, 2021)	Naïve Bayes	Klasifikasi Berita Menggunakan Metode Multinomial Naïve Bayes	Dateset yang digunakan pada penelitian ini menggunakan 10.500 dengan tujuh ketegori	96%

Kesimpulan dari perbandingan penelitian terdahulu dengan penelitian yang penulis lakukan mengenai *klasifikasi Message Spam* dengan menggunakan metode *Naïve Bayes* dan *Multinomial Naïve Bayes* yaitu:

- Jumlah dataset yang digunakan untuk menguji suatu metode sangat mempengaruhi tingkat akurasi. Semakin banyak data yang digunakan maka akan semakin tinggi tingkat akurasinya. Namun jumlah dataset ini harus disesuaikan dengan kebutuhan sistem karena jika tidak sesuai dengan kebutuhan sistem maka sistem nya akan *overfitting*.
- Dalam menentukan klasifikasi, sistem sangat bergantung dengan frekuensi kata pada dataset yang digunakan.

## 2.2 Klasifikasi

Klasifikasi dapat didefinisikan sebagai pengelompokan objek kedalam kelas tertentu berdasarkan kelompoknya yang biasanya disebut dengan kelas (*class*). Salah satu metode klasifikasi yang sering digunakan adalah metode *Naïve Bayes*. Berdasarkan penelitian yang telah dilakukan sebelumnya menggunakan metode *Naïve Bayes*.

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu pembangunan model sebagai memori dan penggunaan model tersebut untuk melakukan klasifikasi pada suatu objek data lain agar diketahui dikelas mana objek data tersebut dalam model yang sudah disimpannya (Putro dkk., 2020).

### **2.3 Message (Pesan)**

Pesan adalah sesuatu yang disampaikan pengirim dengan cara tatap muka atau melalui media komunikasi yang berisi pesan *spam* atau *non-spam* bisa berupa ilmu pengetahuan hiburan, informasi, nasehat atau propaganda. Pesan atau *message* dalam media komunikasi merupakan suatu fitur yang sangat penting. Fitur ini terdapat di beberapa *message platform* yang akan diuraikan pada subbab berikut.

#### **2.3.1 WhatsApp Message**

*WhatsApp Message* adalah salah satu layanan *Instant Messaging* (IM) yang penggunaannya sangat masif dan berkembang sangat pesat hingga saat ini. *WhatsApp Messenger* adalah layanan komunikasi pertukaran pesan di *platform Android* dan *Desktop*. Fitur-fitur dalam aplikasi *WhatsApp* antara lain *chat*, dokumen, gambar, video, lokasi pengguna, dan pesan audio. *WhatsApp* sangat potensial menjadi media kejahatan kriminal terutama kasus penipuan mengingat jumlah pengguna *platform* yang cukup besar dan tersebar.

Kelengkapan fitur yang memaksimalkan pengalaman pemanfaatan teknologi bagi *user* secara umum dapat dimanfaatkan para pelaku kejahatan kriminal untuk memaksimalkan rencana kejahatan. Fitur penghapus pesan hingga penghapusan riwayat panggilan dapat menjadi salah satu fitur yang dimanfaatkan pelaku kejahatan untuk menyembunyikan bukti kejahatan. Oleh karena itu dibutuhkan protokol tertentu yang dapat membantu pihak berwajib untuk mengungkap bukti kejahatan yang telah berusaha dihilangkan oleh pelaku kejahatan (Plianda & Indrayani, 2022).

Berdasarkan uraian di atas, *WhatsApp Message* dapat didefinisikan sebagai aplikasi paling populer dengan jumlah pengguna terbanyak di dunia. *User WhatsApp* dapat memanfaatkan fasilitas seperti mengirim pesan, gambar, video dan *video call*. *WhatsApp* merupakan aplikasi yang paling sering digunakan dengan durasi yang paling lama oleh *user smartphone*.

### 2.3.2 *Short Message Service (SMS)*

*Short Message Service (SMS)* adalah suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat *nirkabel*, yaitu perangkat komunikasi telepon seluler, dalam hal ini perangkat *nirkabel* yang digunakan adalah telepon seluler. Salah satu kelebihan dari SMS adalah biaya yang murah. Selain itu, SMS merupakan metode *store* dan *forward* sehingga keuntungan yang didapatkan adalah pada saat telepon seluler penerima tidak dapat dijangkau, dalam arti tidak aktif atau diluar *service area*.

SMS atau yang dikenal dengan istilah layanan pesan singkat. Layanan pesan singkat ini diciptakan sebagai sarana media komunikasi yang bisa digunakan untuk mengirim pesan secara digital kepada orang yang kita tuju, selayaknya mengirim pesan biasa pengguna SMS harus memiliki alamat yang dituju yaitu menggunakan nomer telepon penggunanya.

SMS merampingkan komunikasi dibanyak tingkatan, diantaranya sebagai berikut :

- Komunikasi cepat: pembaruan singkat antara keluarga dan teman.
- Lansiran: pesan suara, pertanyaan prospek penjualan, janji temu, rapat, atau pengiriman.

Semakin berkembangnya teknologi SMS banyak disalahgunakan oleh pengguna yang tidak bertanggung jawab untuk mengirim pesan-pesan yang tidak bermanfaat dan untuk melakukan tindak kejahatan penipuan. Hal ini sangat mengganggu kenyamanan bagi penerima pesan. Pesan-pesan *spam* banyak beredar mulai dari menawarkan layanan iklan, informasi perbankan dan promosi diskon dari pihak-pihak lain yang mengatas namakan *provider* tertentu untuk melakukan tindak kejahatan penipuan (Munitasari, Santosa, & Supriyanto, 2018).

### 2.3.3 *Email*

*Email* adalah singkatan dari *electronic mail* yang merupakan surat atau pesan dengan format digital. *Email* dapat diakses dengan mudah dengan berbagai *gadget* seperti komputer maupun ponsel *smartphone*.

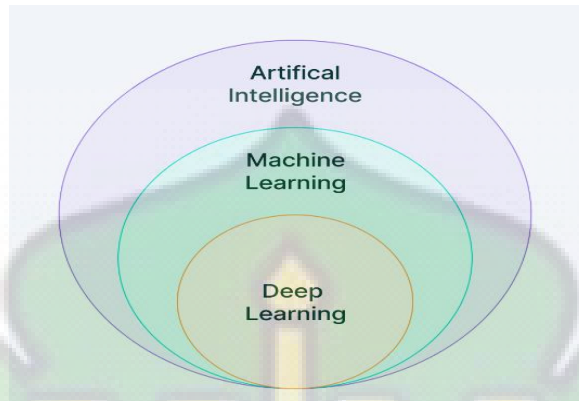
Istilah “*spam*” berawal dari sketsa televisi tahun 70-an berjudul *Monty Python’s Flying Circus*. Dalam salah satu episodenya, sekelompok Viking menyanyikan lagu tentang “*spam*”, yang ternyata merek produk asing olahan. Mereka meyakini pelayan restoran yang sering menawarkan daging “*spam*” meski tak diinginkan pelanggan.

*Spam email* datang dalam berbagai bentuk, yang paling populer adalah untuk mempromosikan penipuan langsung atau skema bisnis yang sah secara marjinal. *Spam* biasanya digunakan untuk mempromosikan akses ke obat-obatan farmasi murah, program berat badan, gelar online, peluang kerja, dan perjudian online. Penipuan uang muka adalah contoh yang terkenal.

## 2.4 *Machine Learning*

*Machine learning* adalah cabang dari bidang ilmu kecerdasan buatan dan menjadi salah satu cara yang dapat digunakan dalam kecerdasan buatan untuk mencapai hasil yang maksimal. *Machine learning* mengandalkan bekerja dengan kumpulan data kecil hingga besar dengan memeriksa dan membandingkan data tersebut untuk menemukan pola umum dan mengeksplorasi perbedaannya. Salah satu penerapan yang umum dari *machine learning* adalah prediksi hasil berdasarkan data yang ada. *Machine learning* mampu mempelajari pola dari kumpulan data yang ada, dan kemudian menerapkannya ke kumpulan data yang tidak diketahui untuk memprediksi hasilnya. Teknik *klasifikasi* adalah teknik yang sering digunakan dalam *machine learning* untuk melakukan suatu proses prediksi hasil (Yunanto dkk., 2021).

*Machine learning* (ML) adalah salah satu perwakilan AI, yang memungkinkan mesin untuk belajar dan memperbaiki secara mandiri (Saluky, 2018).



Gambar 2.1 *Machine Learning* sebagai cabang ilmu *Artificial Intelligence* (AI)  
Sumber: Pragati Baheti

## 2.5 *Natural Language Processing* (NLP)

*Natural language processing* (NLP) adalah metode yang memproses suatu teks menjadi kata kunci bagi *user*. *Natural language processing* merupakan cabang dari ilmu komputer dan *linguistik* yang membahas tentang interaksi antara manusia dengan komputer menggunakan bahasa alami atau bahasa manusia (Migunani & Kevin Aditama, 2020).

Tujuan utama NLP adalah untuk digunakan dalam berbagai tugas atau aplikasi yang melibatkan bahasa manusia. Dalam konteks tersebut, sistem NLP mewakili arti dan maksud yang sebenarnya dari permintaan pengguna. Sistem NLP digunakan untuk memahami permintaan yang diekspresikan secara alami dalam bahasa sehari-hari, seolah-olah pengguna sedang berbicara dengan seorang pustakawan referensi. Dengan menggunakan teknik-teknik NLP, sistem dapat menganalisis dan menginterpretasi permintaan pengguna dengan akurasi dan memprosesnya untuk memberikan respons yang sesuai.

## 2.6 *TensorFlow*

*TensorFlow* adalah *library open source* yang dapat diterapkan oleh *machine learning* dengan skala besar dan dikembangkan oleh google. *TensorFlow* merupakan lanjutan dari *DistBelief* yang telah diliris oleh google

sebelumnya dan memiliki kemampuan melatih model besar dengan memanfaatkan kluster komputasi dengan ribuan mesin (Juli dkk., 2023).

*TensorFlow* menyediakan antarmuka untuk mengekspresikan algoritma *machine learning* secara *fleksibel* dan dapat berjalan diberbagai sistem. *TensorFlow* merupakan salah satu *library* untuk data *science* paling populer dengan memiliki jumlah pengembang dan komunitas yang besar (Ihsan, 2021).

## 2.7 *Naïve Bayes*

*Naïve Bayes* adalah suatu proses klasifikasi dengan probabilitas sederhana yang mengacu pada teori *bayes*. Teori *bayes* menyatakan bahwa kemungkinan terjadinya suatu peristiwa sama dengan probabilitas intrinsik dikalikan probabilitas bahwa hal serupa akan terjadi lagi di masa depan. *Naïve Bayes* adalah algoritma pembelajaran probabilitas yang berasal dari teori keputusan *bayesian* (Mustofa & Mahfudh, 2019).

Metode *Naïve Bayes* memiliki beberapa kelebihan yaitu :

- Kesederhanaan dalam proses komputerisasi.
- Nilai akurasi yang tinggi.
- Kecepatan dalam memproses dataset

*Naïve bayes* memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya. Pada proses *training*, dokumen input telah diketahui kelasnya. Proses *training* berguna untuk membentuk pengetahuan berupa nilai probabilitas kata. Perlu diketahui bahwa pada proses *training*, tidak dijalankan modul klasifikasi, tetapi hanya menghasilkan dokumen yang mengandung kata untuk mengkararakteristik suatu kelas (Haryono dkk., 2018).

Metode *Naïve Bayes* memiliki beberapa tahapan :

1. Pelatihan
  - a. Menghitung probabilitas kelas
  - b. Menghitung probabilitas kata terhadap setiap kelas
2. Pengklasifikasian
  - a. Menghitung probabilitas dokumen terhadap setiap kelas
  - b. Menentukan kelas dengan nilai probabilitas yang paling tinggi



Persamaan dari teorema *bayes* adalah :

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \quad (2.1)$$

Keterangan :

- $X$  : Data dengan kelas yang belum diketahui
- $H$  : Hipotesis data  $X$  merupakan suatu kelas spesifik
- $P(H|X)$  : Probabilitas hipotesis  $H$  berdasarkan kondisi  $X$
- $P(H)$  : Probabilitas hipotesis  $H$
- $P(X|H)$  : Probabilitas  $X$  berdasarkan kondisi pada hipotesis  $H$
- $P(X)$  : Probabilitas  $X$

Untuk menjelaskan teorema *Naïve Bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut (Aripin, 2019).

## 2.8 *Multinomial Naïve Bayes*

*Multinomial Naïve Bayes* merupakan variasi dari *Naïve Bayes* yang digunakan untuk mengklasifikasi dokumen atau teks ke dalam beberapa kategori atau kelas berdasarkan fitur-fitur yang ada dalam dokumen. Metode ini melakukan klasifikasi dengan pembelajaran *supervised* menggunakan model probabilistik. *Multinomial Naïve Bayes* dipengaruhi oleh serangkaian *term*, dengan kata lain jumlah *term* diperhitungkan. Peluang antara *term* satu dengan yang lain tidak bergantung. Model *Multinomial Naïve Bayes* memperhitungkan frekuensi setiap kata yang muncul pada dokumen (Dewi, 2021).

Metode *Multinomial Naïve Bayes* memanfaatkan teorema probabilitas yaitu teorema *bayes* dan fungsionalitas *data mining* yaitu klasifikasi *naïve bayesian*. *Multinomial Naïve Bayes* adalah salah satu metode *bayes* yang digunakan dengan memperhitungkan frekuensi masing-masing kemunculan data dalam sebuah dokumen dan probabilitas. Kelebihan *Multinomial Naïve Bayes* diantaranya adalah tingkat akurasi yang tinggi, mudah diimplementasikan, waktu komputasi yang rendah serta *error rate* yang minimum (Sriyano & Setiawan, 2021).

## 2.9 Confusion Matrix

*Confusion Matrix* adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah (Normawati & Prayogi, 2021). *Confusion Matrix* sering digunakan dalam *machine learning* untuk mengukur kinerja model klasifikasi.

*Confusion matrix* merupakan suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data *mining*. Terdapat empat istilah representasi hasil klasifikasi, yaitu *True Positif* (TP) merupakan nilai positif yang dideteksi dengan benar, *True Negatif* (TN) merupakan jumlah data negatif yang dideteksi dengan benar, *False Positif* (FP) merupakan data negatif namun terdeteksi secara positif dan *False Negatif* (FN) merupakan data negatif terdeteksi sebagai data negatif (Mutawalli dkk., 2019).

Table 2.2 *Confusion Matrix*

<i>Confusion Matrix</i>	Prediksi kelas	
	Negatif	Positif
Negatif	TN	FN
Positif	FP	TP

Keterangan :

1. TP (*True Positive*) = jumlah dokumen dari kelas 1 yang benar diklasifikasikan sebagai kelas 1. *True Positive* (TP) ini adalah jumlah instance yang benar-benar termasuk dalam kelas *positive* dan juga diprediksi dengan benar oleh model sebagai kelas *positive*. Dalam kata lain, model berhasil mengidentifikasi *instance positive* dengan benar.
2. TN (*True Negative*) = jumlah dokumen dari kelas 0 yang benar diklasifikasikan sebagai kelas 0. *True Negative* (TN) ini adalah jumlah instance yang benar-benar termasuk dalam kelas *negative* dan juga diprediksi dengan benar oleh model sebagai kelas *negative*. Model berhasil mengidentifikasi *instance negative* dengan benar.
3. FP (*False Positive*) = jumlah dokumen dari kelas 0 yang salah diklasifikasikan sebagai kelas 1. *False Positive* (FP) Ini adalah jumlah instance yang sebenarnya termasuk dalam kelas *negative*, tetapi diprediksi

oleh model sebagai kelas *positive*. Kesalahan ini juga dikenal sebagai "*Type I error*" atau "*False Alarm*". Model salah memprediksi *instance negative* sebagai *positive*.

4. FN (*False Negative*) = jumlah dokumen dari kelas 1 yang salah diklasifikasikan sebagai kelas 0. *False Negative* (FN) Ini adalah jumlah *instance* yang sebenarnya termasuk dalam kelas *positive*, tetapi diprediksi oleh model sebagai kelas *negative*. Kesalahan ini juga dikenal sebagai "*Type II error*" atau "*Miss*". Model gagal mengidentifikasi *instance positive*.

Rumus *confusion matrix* untuk menghitung *accuracy*, *precision*, dan *recall* dapat dilihat pada persamaan 2.2, 2.3, 2.4 dan 2.5.

$$Accuracy = \frac{TP+TN}{Total} * 100\% \quad (2.2)$$

$$Precision = \frac{TP}{TP+FP} * 100\% \quad (2.3)$$

$$Recall = \frac{TP}{TP+FN} * 100\% \quad (2.4)$$

$$F1-Score = \frac{2*precision*recall}{precision+recall} * 100\% \quad (2.5)$$

Keterangan :

- *Accuracy* merupakan hasil perhitungan tingkat keakuratan deteksi objek terhadap objek dataset secara keseluruhan.
- *Precision* merupakan persamaan mengenai jumlah prediksi yang benar dibandingkan dari keseluruhan hasil yang dapat diprediksi oleh sistem. *Precision* juga dapat menebak dan membedakan mana *spam* dan *non-spam*.
- *Recall* merupakan jumlah prediksi yang benar dibandingkan dengan keseluruhan hasil pendeteksian dataset sebenarnya. Persamaan ini akan menghasilkan jumlah data yang terdeteksi dengan benar dari keseluruhan jumlah data yang terdeteksi oleh sistem.

- *F1-Score* untuk menginformasikan nilai dari keduanya yaitu dengan cara mengkombinasikan nilai *precision* dan *recall*.

## **2.10 Tools**

Penelitian ini *tools* yang digunakan oleh penulis adalah bahasa pemrograman *Python* dan *Google Colaboratory*. *Tools* ini dapat digunakan secara online dan tidak berbayar pada *Website Google Colaboratory*.

### **2.10.1 Python**

*Python* adalah salah satu bahasa pemrograman tingkat tinggi (*high-level programming language*), berjalan dengan sistem *interpreted* dan bisa dipakai untuk berbagai tujuan (*general purpose*). *Python* diciptakan oleh Guido van Rossum pertama kali di *Scitching Mathematisch Centrum (CWI)* di Belanda pada awal tahun 1990-an (Gumilar dkk., n.d.).

*Python* secara umum berbentuk pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Istilah lainnya, bahasa pemrograman *multi-paradigma*. *Python* juga dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai *platform* sistem operasi (Enterprise, 2019).

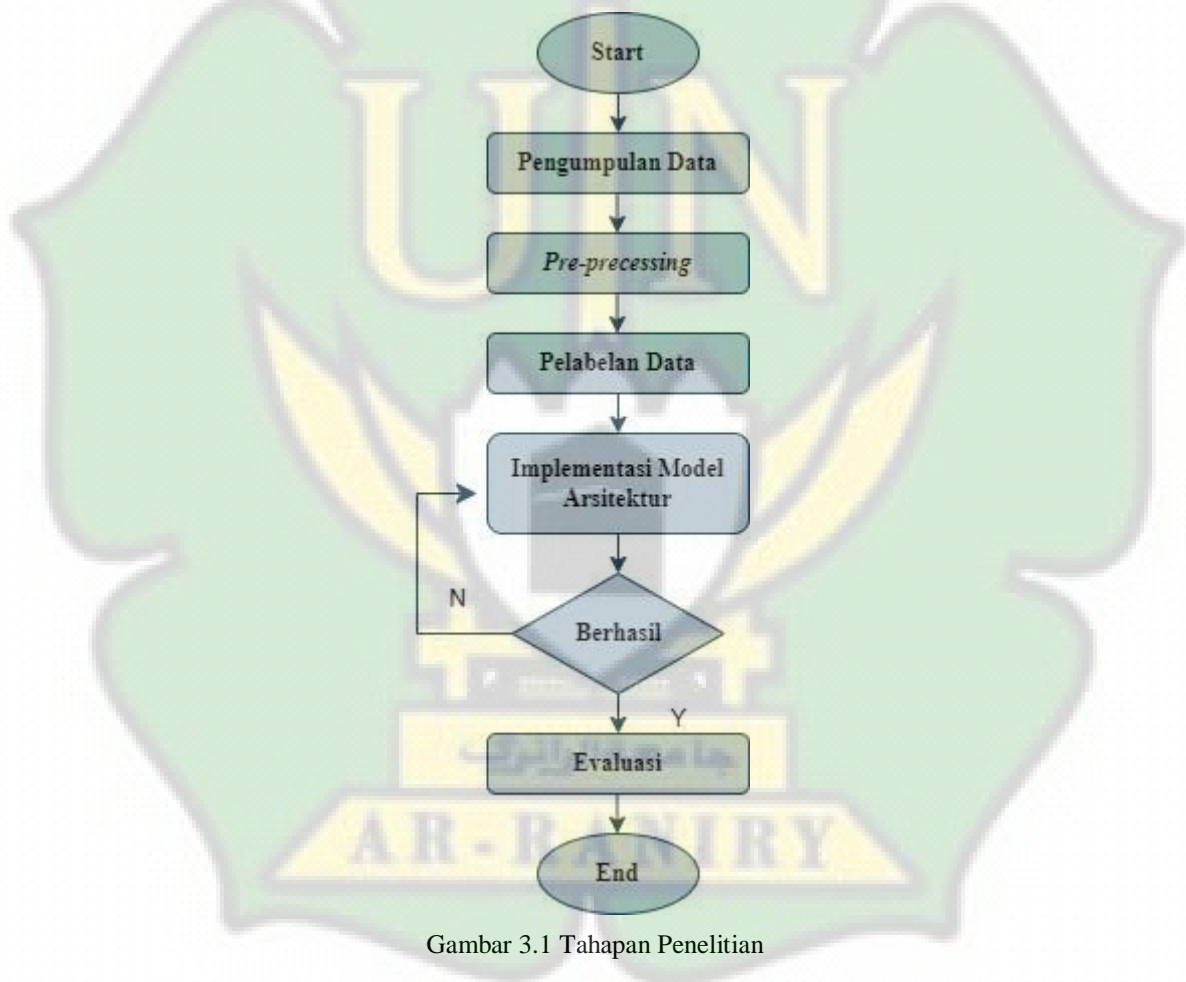
### **2.10.2 Google Colaboratory**

*Google Colaboratory* atau “*Colab*” merupakan produk dari *Google Research*. *Colab* memungkinkan siapa saja menulis dan mengeksekusi kode *python* *arbitrer* melalui *browser*, dan sangat cocok untuk *machine learning*, analisis data, serta pendidikan. Secara lebih teknis, *Colab* merupakan layanan *notebook Jupyter* yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke *resource* komputasi termasuk GPU. *Resource Colab* tidak dijamin dan sifatnya terbatas, serta batas penggunaannya terkadang berfluktuasi. Hal ini diperlukan agar *Colab* dapat menyediakan *resource* secara gratis (Soen dkk., 2022).

## BAB III METODE PENELITIAN

### 3.1 Tahapan Penelitian

Bagian ini penulis menjelaskan tahapan yang akan dilakukan dalam penelitian. Tahapan penelitian ini bertujuan untuk memudahkan peneliti dalam melakukan klasifikasi *Message Spam*. Berikut tahapan penelitian yang akan dilakukan dapat dilihat pada gambar 3.1.



### 3.2 Pengumpulan Data

Sumber data yang digunakan oleh penulis dalam penelitian ini adalah data *WhatsApp*, *SMS*, dan *Email*. Pengumpulan data didapatkan dari pesan *WhatsApp* yang dikumpulkan secara manual sedangkan data *SMS* dan *Email* diperoleh dari *Github.com*

Jumlah dataset yang digunakan pada penelitian ini sebanyak 11.801 yang dikumpulkan dari tiga *platform* yang berbeda. Uraian dataset ini yaitu terdiri dari 500 pesan *WhatsApp*, 5565 dari *SMS* dan *email* sebanyak 5736. Pesan-pesan ini akan diklasifikasikan menjadi dua kategori, yaitu *spam* dan *non-spam*.

### 3.3 Pre-processing Data

Berikut ini beberapa tahap *pre-processing* umum yang biasanya dilakukan sebelum menggunakan *Naive Bayes* dalam klasifikasi teks:

- *Case Folding*: Data teks seringkali mengandung karakter khusus, seperti tanda baca, simbol, atau karakter tidak diinginkan lainnya. Tahap *pre-processing* ini melibatkan penghapusan karakter khusus tersebut agar tidak mempengaruhi analisis dan klasifikasi.

Tabel 3.1 Contoh *Case Folding*

Contoh Teks Awal	Output Hasil <i>Case Folding</i>
Happy Weekend BB Lovers. Sabtu & Minggu ini kita ada kejutan menarik dari Outlet @kosngosan X @kosngosan loh! Promo fantastis yang ditunggu-tunggu akhirnya dateng lag nihh!!! Yuk sore ini kita ketemu di Outlet Medan yaa, waktu nya terbatas bebs! 20-21 Agustus 2022. 15.00 – 20.00 WIB.	Happy Weekend BB Lovers. Sabtu & Minggu ini kita ada kejutan menarik dari Outlet kosngosan X kosngosan loh Promo fantastis yang ditunggu-tunggu akhirnya dateng lag nihh Yuk sore ini kita ketemu di Outlet Medan yaa, waktu nya terbatas bebs 20-21 Agustus 2022. 15.00 – 20.00 WIB.

- *Tokenisasi*: *Tokenisasi* adalah proses memecah teks menjadi unit-unit terpisah, seperti kata-kata atau frasa. Ini penting dalam mempersiapkan data teks untuk analisis lebih lanjut. Biasanya, teks dipisahkan berdasarkan spasi,

namun dalam beberapa kasus, tokenisasi yang lebih canggih seperti tokenisasi menggunakan aturan *grammar* atau pemisahan frasa juga dapat diterapkan.

Tabel 3.2 Contoh Tokenisasi

Contoh Teks Awal	Output Hasil Tokenisasi
Siapin pertanyaan terus masing2 yaa	['Siapin', 'pertanyaan', 'terus', 'masing2', 'yaa']

- Mengubah menjadi *lowercase*: Mengubah semua huruf dalam teks menjadi *lowercase* dapat membantu menghindari duplikasi kata-kata yang memiliki huruf besar/kecil berbeda dan membantu memperlakukan kata-kata yang sama dengan cara yang konsisten.

Tabel 3.3 Contoh Mengubah Menjadi *Lowercase*

Contoh Teks Awal	Output Hasil <i>Lowercase</i>
INFORMASI SHOPEE. SPECIAL HARI INI PROGRAM PROMO PUNCAK SHOPEE BIG RAMADHAN SALE. Pemberitahuan Mohon maaf mengganggu waktu nya. Saya EDI HERMAWAN dari PT SHOPEE IND TBK. Kami ucapkan selamat, untuk KK sebagai member shopee atau nomor ponsel yang telah terpilih.	Informasi shopee. Special hari ini program promo puncak sh0pee big ramadhan sale. Pemberitahuan mohon maaf mengganggu waktu nya. Saya edi hermawan dari pt shopee ind tbk. kami ucapkan selamat, untuk kk sebagai member shopee atau nomor ponsel yang telah terpilih.

- Menghilangkan *stopwords*: *Stopwords* adalah kata-kata umum yang tidak memberikan informasi signifikan untuk analisis, seperti "a", "the", "is", "and", dan sebagainya. Menghilangkan *stopwords* dapat membantu mengurangi dimensi fitur dan meningkatkan efisiensi klasifikasi.

Tabel 3.4 Contoh Menghilangkan *Stopwords*

Contoh Teks Awal	Ouput Hasil <i>Stopwords</i>
Apakah klen jam 9 an besok ada yg free Mau minta kawanin ke	jam 9 free minta kawanin pabrik tahu

pabrik tahu	
-------------	--

- *Stemming* atau Lemmatisasi: *Stemming* dan lemmatisasi adalah proses untuk mengubah kata-kata menjadi bentuk akar atau kata dasar mereka. Tujuannya adalah untuk mengurangi variasi morfologis dalam kata-kata yang memiliki akar yang sama, sehingga kata-kata dengan akar yang sama dapat diperlakukan sebagai fitur yang serupa dalam klasifikasi.

Tabel 3.5 Contoh *Stemming*

Contoh Teks Awal	Output Hasil <i>Stemming</i>
yang mau ikut praktek silakan di amankan yaa	yang mau ikut praktik silakan di amankan yaa

Tahap *pre-processing* ini membantu membersihkan dan mempersiapkan data teks sebelum diterapkan pada metode *Naive Bayes* untuk klasifikasi. *Pre-processing* yang tepat dapat membantu meningkatkan kualitas dan kinerja model klasifikasi.

### 3.4 Pelabelan Data

Pada penelitian ini akan dilakukan pelabelan pada message yang sudah dikumpulkan dalam data *WhatsApp*, *SMS*, dan *Email* sehingga terdiri dari label *spam* dan *non-spam*.

Teknik pelabelan data menggunakan metode manual yang dilakukan dengan menentukan *spam* dan *non-spam* pada data. Cara menentukan label *spam* dan *non-spam* yaitu dengan melihat jenis pesan yang masuk. Untuk pesan yang dikategorikan *non-spam* berupa pesan promosi tidak diminta, pesan tidak dikenali, pesan mengandung tautan atau permintaan data, dan pesan tidak relevan. Contoh kalimat yang sudah ditentukan lebelnya dapat dilihat pada tabel 3.1.

Tabel 3.6 Contoh Pelabelan Kalimat

No	Kalimat	Label
1	Pelanggan Telkomsel Yth di mana pun Anda berada, Terima kasih ya, tetap setia menggunakan produk dan layanan Telkomsel. Telkomsel menghimbau agar Bapak/Ibu selalu	<i>Spam</i>

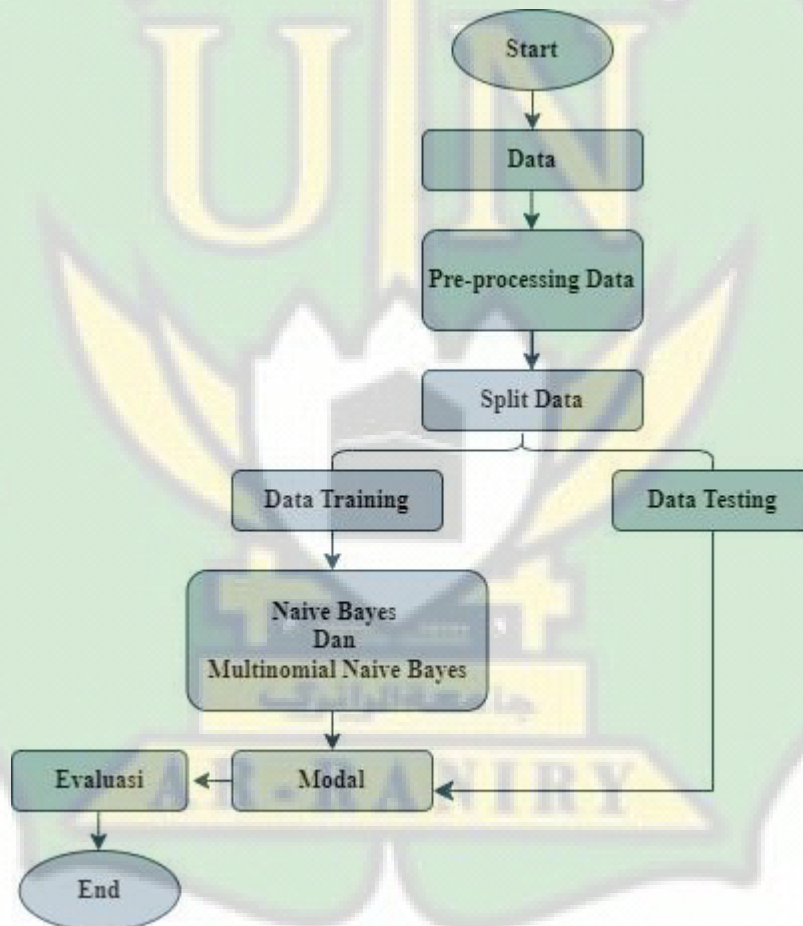


No	Kalimat	Label
	<p>waspada thdp modus penipuan dgn clk file. APK yg berisi undangan pernikahan, lowongan pekerjaan hingga surat tilang, dan lainnya. Cek validasi pengiriman pesannya, jgn langsung meng-klik tautan, dan mengunduh aplikasi sembarangan yang tdk dikenali. Aplikasi MyTelkomsel dpt didownload di <a href="https://tsel.id/downloadmytelkomsel">https://tsel.id/downloadmytelkomsel</a></p>	
2	<p>KHUSUS UNTUK KAKAK KESEMPATAN KEMENANGAN Langsung dari SBOBET 95% Profit &amp; BONUS 200% No Hoax... Cukup Deposit 100k saja saldo jadi 300k. Link web : <a href="http://betmaingold.com">betmaingold.com</a></p>	<i>Spam</i>
3	<p>Guys Besok ada yg free ga siang sore? Mu minta tolong masuk ngajar sehari</p>	<i>Non-Spam</i>
4	<p>Hai Izzia Khalkia, Khusus buat kamu yang transaksi finansial melalui BSI Mobile, dapatkan kesempatan bawa pulang hadiah dari program Hujan Rezeki BSI Mobile 2022. Perbanyak transaksi *Transfer ke Bank Lain dengan metode BI Fast di BSI Mobile, dapatkan 2 poin/trx dan CASHBACK biaya transfer BI Fast sebanyak 4x. Periode program: 05 s.d 31 Desember 2022. Nikmati kemudahan bertransaksi di BSI Mobile dan dapatkan keuntungannya! SnK berlaku. Info: <a href="http://bankbsi.id/HujanRezekiBSIMobile">bankbsi.id/HujanRezekiBSIMobile</a> . #BSIMobile #LifeWithBSI</p>	<i>Spam</i>
5	<p>Assalamu'alaikum maaf ya merepotkan, ini nomor dina yang baru. Hp dina hilang kemarin, kartunya pun udah gak bisa diaktifkan lagi di grapari</p>	<i>Non-Spam</i>
6	<p>wkwk awalnya pun aku gituu jugaa asal jumpa aku jail bgt ampun tapi pas di chat beda bgt parah wkwk</p>	<i>Non-Spam</i>
7	<p>Selamat siang Cut Sari Ramadhani, kakak mau konfirmasi jadinya mau daftar english academy yang paket 1 tahun atau 9 bulan ya? Untuk promo 25% terakhir di hari ini. Masih tersedia</p>	<i>Spam</i>

No	Kalimat	Label
	2 seat dan klaim hadiahnya. Minggu depan sudah mulai kelasnya. Mohon dibalas yaa	

### 3.5 Implementasi Metode

Alur metode *Naive Bayes* dan *Multinomial Naive Bayes* memiliki kesamaan secara umum, karena metode *Multinomial Naive Bayes* adalah varian dari metode *Naive Bayes*. Namun, ada perbedaan dalam langkah-langkah tertentu tergantung pada jenis data yang digunakan. Pada gambar 3.2 merupakan alur sistem pada implementasi metode *Naive Bayes* dan *Multinomial Naive Bayes*.



Gambar 3.2 Alur Sistem Pada Implemen Metode

Keterangan dari alur di atas sebagai berikut :

1. Dataset yang penulis gunakan pada penelitian ini dikumpulkan dari tiga dataset yang berbeda yaitu *Email*, *WhatsApp* dan *SMS*. Jumlah total dataset yang digunakan 11.801 yang terdiri dari dataset *Email* 5736 data yang diperoleh dari github <https://github.com/arshiram/Email-Spam.csv>. Dataset *WhatsApp* terdiri dari 500 data yang dikumpulkan secara manual dan dataset *SMS* yang berjumlah dari 5565 yang diperoleh dari github <https://github.com/mohitgupta-omg/Kaggle-SMS-Spam-Collection-Dataset/blob/master/spam.csv>. Seluruh dataset yang telah dikumpulkan kemudian di uji untuk diklasifikasikan menjadi dua kategori yaitu *spam* dan *non-spam*. Klasifikasi ini adalah komponen penting dalam penelitian ini yang digunakan untuk menganalisis dan mengembangkan model klasifikasi. Data yang sudah dikumpulkan akan dilakukan proses *splitting* data menjadi data *training* dan data *testing* dengan perbandingan 80:20.
2. *Pre-processing* data merupakan proses untuk pembersihan data yang akan digunakan untuk proses analisis. Tahapan yang dilakukan dalam *preprocessing* data yaitu proses *case folding*, *tokenized*, *stopword removed* dan *stemming*.
3. *Split Data*. Data yang sudah dikumpulkan selanjutnya akan dibagi menjadi data *training* dan data *testing* dengan perbandingan 80:20.
4. Data *training* adalah data yang digunakan untuk melatih model. Data *training* diambil dari 80% data pada setiap sumber yang berbeda, yaitu data *WhatsApp*, *Email* dan *SMS*. Pada data *WhatsApp* diambil sebanyak 400 data. Pada data *Email* diambil sebanyak 4585. Dan pada data *SMS* diambil sebanyak 4457 data. Sehingga total data *training* yang digunakan sebanyak 9442 data.
5. Metode yang digunakan pada penelitian ini adalah *Naïve Bayes* dan *Multinomial Naïve Bayes*. *Naïve Bayes* dan *Multinomial Naïve Bayes* adalah algoritma klasifikasi yang digunakan dalam pemrosesan bahasa alami dan analisis teks. Berikut ini merupakan penjelasan perbedaan metode *Naïve Bayes* dan *Multinomial Naïve Bayes*:

**a. Metode *Naïve Bayes***

- Distribusi probabilitas metode *Naïve Bayes* adalah pendekatan statistik yang didasarkan pada teorema *bayes*. Ini digunakan untuk memodelkan distribusi probabilitas dari suatu data berdasarkan data pelatihan.
- Asumsi independensi pada metode ini mengandalkan asumsi bahwa atribut atau fitur dalam data adalah independen satu sama lain. Ini berarti bahwa tidak ada ketergantungan antara atribut, yang disebut sebagai asumsi “*naïve*”. Asumsi ini bisa menjadi terlalu simplistik dalam banyak kasus.
- Aplikasi pada data teks, meskipun *Naïve Bayes* dapat digunakan untuk klasifikasi teks, asumsi independen atribut sering tidak terpenuhi dalam data teks yang mengandung kata-kata atau fitur yang berkaitan erat satu sama lain.

**b. Metode *Multinomial Naïve Bayes***

- Pendekatan khusus untuk data teks, *Multinomial Naïve Bayes* adalah variasi dari *Naïve Bayes* yang secara khusus cocok untuk klasifikasi teks. Ini mempertimbangkan atribut sebagai jumlah kemunculan kata-kata dalam dokumen, dan oleh karena itu, menggantikan asumsi independensi dengan asumsi bahwa atribut ini mengikuti distribusi *multinomial*.
  - Perhitungan berdasarkan jumlah kata dalam *Multinomial Naïve Bayes*, frekuensi kemunculan kata dalam dokumen adalah atribut penting. Model ini menggunakan informasi berapa kali kata tertentu muncul dalam dokumen sebagai dasar untuk menghitung probabilitas kelas.
  - Tidak memperhatikan urutan dan bobot kata, metode *Multinomial Naïve Bayes* tidak memperhitungkan urutan kata atau bobot kata. Setiap kata diperlakukan dengan nilai yang setara dalam klasifikasi.
6. Model adalah representasi matematis yang memungkinkan perhitungan probabilitas yang diperlukan untuk mengklasifikasikan data. Pada tahap ini, model akan dikembangkan dengan menerapkan dua metode, yaitu *Naïve Bayes* dan *Multinomial Naïve Bayes*. Model akan dilatih dengan menggunakan data *training*.
7. Data *testing* adalah data yang digunakan untuk menguji performa model yang telah dilatih. Data *testing* diambil dari 20% data pada setiap sumber yang

berbeda, yaitu data *WhatsApp*, *Email* dan *SMS*. Pada data *WhatsApp* diambil sebanyak 100 data. Pada data *Email* diambil sebanyak 1147. Dan pada data *SMS* diambil sebanyak 1115 data. Jumlah total data *testing* yang digunakan sebanyak 2362 data.

8. Evaluasi merupakan tahap penting untuk mengukur sejauh mana model klasifikasi berhasil dilakukan. Pada tahap evaluasi menggunakan *confusion matrix* seperti akurasi, presisi, *recall* dan *f1-score* untuk memeriksa kinerja model yang dibangun. Pada tahap evaluasi data yang dipakai yaitu data *testing* yang diambil dari 20% data pada setiap sumber yang berbeda, yaitu data *WhatsApp*, *Email* dan *SMS*.

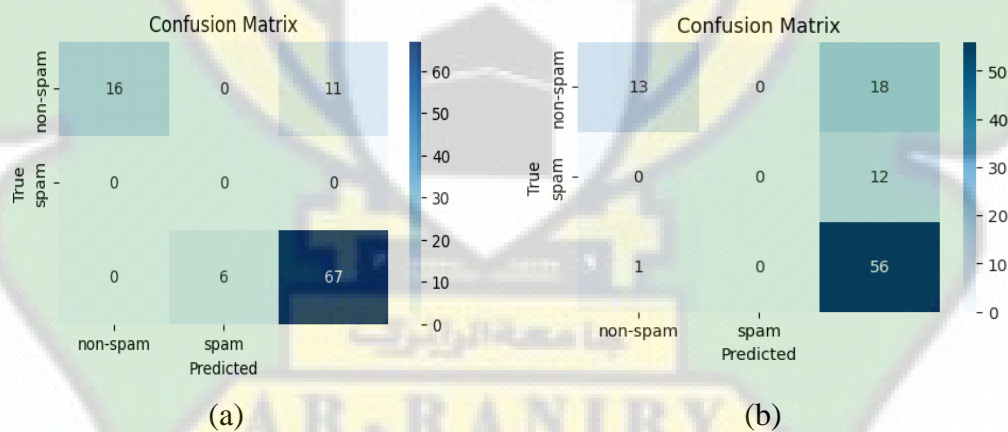


## BAB IV HASIL DAN PEMBAHASAN

Pada bab ini, penulis akan menjelaskan tentang proses klasifikasi *message spam* dengan menggunakan metode *Naïve Bayes* dan *Multinomial Naïve Bayes* untuk mengetahui tingkat akurasi kedua metode tersebut. Pada penelitian ini penulis akan menjelaskan proses klasifikasi pesan *spam* dan *non-spam* dari implementasi metode *Naïve Bayes* dan *Multinomial Naïve Bayes*.

### 4.1 Hasil Uji Coba *Confusion Matrix*

Setelah dilakukan proses training data pada model *Naïve Bayes* dan *Multinomial Naïve bayes*, maka selanjutnya akan dilakukan uji coba untuk mengetahui performa model terhadap data *WhatsApp*, *SMS* dan *Email*. Diagram *confusion matrix* dan performa model dari data *WhatsApp* dapat dilihat pada gambar 4.1 dibawah ini.



Gambar 4.1 Hasil Akurasi *Confusion Matrix* dari Dataset *WhatsApp*

Gambar (a) merupakan hasil *confusion matrix* dari dataset *WhatsApp* dengan menggunakan metode *Naïve Bayes*, gambar (b) merupakan hasil *confusion matrix* dari dataset *WhatsApp* dengan menggunakan metode *Multinomial Naïve Bayes*. Tabel perhitungan untuk mencari nilai TP, TN, FP dan FN untuk metode *Naïve Bayes* dan *Multinomial Naïve Bayes* dapat dilihat pada tabel 4.1 dibawah ini.

Tabel 4.1 Perhitungan *Confusion Matrix* menggunakan dataset *WhatsApp*

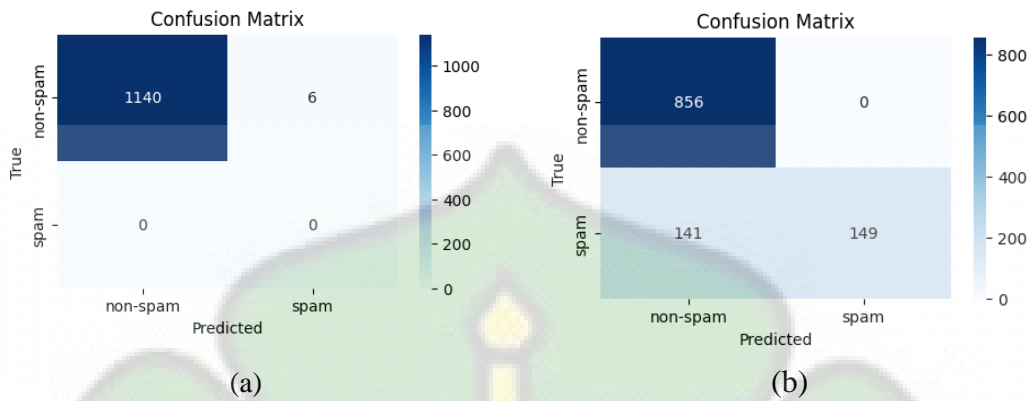
Nilai	Metode	
	Naïve Bayes	Multinomial Naïve Bayes
<i>True Positive</i>	16	13
<i>True Negative</i>	$0 + 0 + 6 + 67 = 73$	$0 + 12 + 0 + 56 = 68$
<i>False Positive</i>	$0 + 0 = 0$	$0 + 1 = 1$
<i>False Negative</i>	$0 + 11 = 11$	$0 + 18 = 18$

Perhitungan untuk mencari nilai akurasi, *recall*, *precision* dan *F1-Score* dari dataset *WhatsApp* dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil akurasi, *recall*, *precision* dan *f1-score* dari Dataset *WhatsApp*

Metode	
Naïve Bayes	Multinomial Naïve Bayes
$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} * 100\%$ $= \frac{16+73}{16+73+0+11} * 100\%$ $= \frac{89}{100} * 100\%$ $= \mathbf{89\%}$	$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} * 100\%$ $= \frac{13+68}{13+68+1+18} * 100\%$ $= \frac{81}{100} * 100\%$ $= \mathbf{81\%}$
$\text{Recall} = \frac{TP}{TP+FN} * 100\%$ $= \frac{16}{16+11} * 100\%$ $= \mathbf{59\%}$	$\text{Recall} = \frac{TP}{TP+FN} * 100\%$ $= \frac{13}{13+18} * 100\%$ $= \mathbf{41\%}$
$\text{Precision} = \frac{TP}{TP+FP} * 100\%$ $= \frac{16}{16+0} * 100\%$ $= \mathbf{100\%}$	$\text{Precision} = \frac{TP}{TP+FP} * 100\%$ $= \frac{13}{13+1} * 100\%$ $= \mathbf{92\%}$
$\text{F1-Score} = \frac{2*\text{recall}*\text{precision}}{\text{recall}+\text{precision}} * 100\%$ $= \frac{2*59*100}{59+100} * 100\%$ $= \mathbf{74\%}$	$\text{F1-Score} = \frac{2*\text{recall}*\text{precision}}{\text{recall}+\text{precision}} * 100\%$ $= \frac{2*41*92}{41+92} * 100\%$ $= \mathbf{56\%}$

Diagram *confusion matrix* dan performa model dari data *Email* dapat dilihat pada gambar 4.2 dibawah ini.



Gambar 4.2 Hasil Akurasi *Confusion Matrix* dari Dataset *Email*

Gambar (a) merupakan hasil *confusion matrix* dari dataset *Email* dengan menggunakan metode *Naïve Bayes*, gambar (b) merupakan hasil *confusion matrix* dari dataset *Email* dengan menggunakan metode *Multinomial Naïve Bayes*. Tabel perhitungan untuk mencari nilai TP, TN, FP dan FN untuk metode *Naïve Bayes* dan *Multinomial Naïve Bayes* dapat dilihat pada tabel 4.3 dibawah ini.

Tabel 4.3 Perhitungan *Confusion Matrix* menggunakan dataset *Email*

Nilai	Metode	
	Naïve Bayes	Multinomial Naïve Bayes
<i>True Positive</i>	1140	856
<i>True Negative</i>	0	149
<i>False Positive</i>	0	141
<i>False Negative</i>	6	0

Perhitungan untuk mencari nilai akurasi, *recall*, *precision* dan *F1-Score* dari dataset *Email* dapat dilihat pada tabel 4.4.

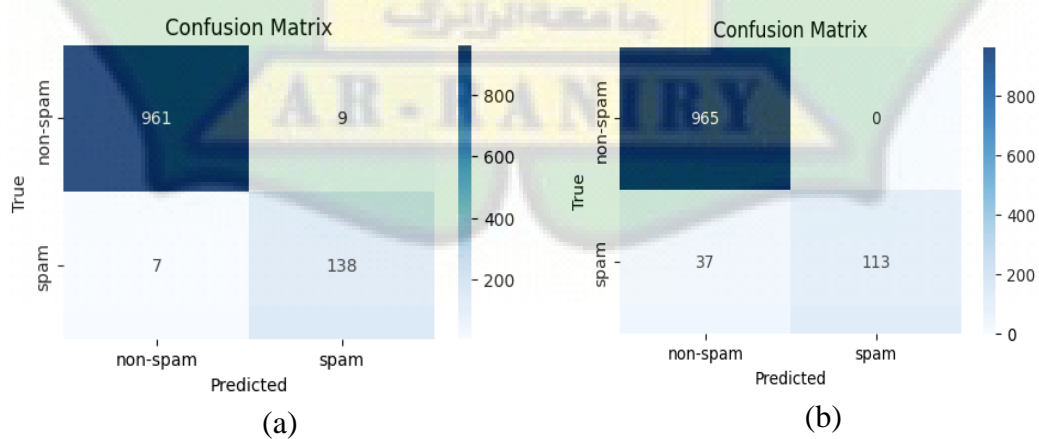
Tabel 4.4 Hasil akurasi, *recall*, *precision* dan *f1-score* dari Dataset *Email*

Metode	
Naïve Bayes	Multinomial Naïve Bayes
$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\%$	$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\%$



Metode	
Naïve Bayes	Multinomial Naïve Bayes
$= \frac{1140+0}{1140+0+0+6} * 100\%$ $= \frac{1140}{1146} * 100\%$ $= \mathbf{99\%}$	$= \frac{856+149}{856+149+141+0} * 100\%$ $= \frac{1005}{1146} * 100\%$ $= \mathbf{87\%}$
$Recall = \frac{TP}{TP+FN} * 100\%$ $= \frac{1140}{1140+6} * 100\%$ $= \mathbf{99\%}$	$Recall = \frac{TP}{TP+FN} * 100\%$ $= \frac{856}{856+0} * 100\%$ $= \mathbf{100\%}$
$Precision = \frac{TP}{TP+FP} * 100\%$ $= \frac{1140}{1140+0} * 100\%$ $= \mathbf{100\%}$	$Precision = \frac{TP}{TP+FP} * 100\%$ $= \frac{856}{856+141} * 100\%$ $= \mathbf{85\%}$
$F1-Score = \frac{2*recall*precision}{recall+precision} * 100\%$ $= \frac{2*99*100}{99+100} * 100\%$ $= \mathbf{99\%}$	$F1-Score = \frac{2*recall*precision}{recall+precision} * 100\%$ $= \frac{2*100*85}{100+85} * 100\%$ $= \mathbf{91\%}$

Diagram *confusion matrix* dan performa model dari data SMS dapat dilihat pada gambar 4.3 dibawah ini.



Gambar 4.3 Hasil Akurasi *Confusion Matrix* dari Dataset SMS

Gambar (a) merupakan hasil *confusion matrix* dari dataset SMS dengan menggunakan metode *Naïve Bayes*, gambar (b) merupakan hasil *confusion matrix* dari dataset SMS dengan menggunakan metode *Multinomial Naïve Bayes*. Tabel perhitungan untuk mencari nilai TP, TN, FP dan FN untuk metode *Naïve Bayes* dan *Multinomial Naïve Bayes* dapat dilihat pada tabel 4.5 dibawah ini.

Tabel 4.5 Perhitungan *Confusion Matrix* menggunakan dataset SMS

Nilai	Metode	
	Naïve Bayes	Multinomial Naïve Bayes
<i>True Positive</i>	961	965
<i>True Negative</i>	138	113
<i>False Positive</i>	7	37
<i>False Negative</i>	9	0

Perhitungan untuk mencari nilai akurasi, *recall*, *precision* dan *F1-Score* dari dataset SMS dapat dilihat pada tabel 4.6.

Tabel 4.6 Hasil akurasi, *recall*, *precision* dan *f1-score* dari Dataset SMS

Metode	
Naïve Bayes	Multinomial Naïve Bayes
$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} * 100\%$ $= \frac{961+138}{961+138+7+9} * 100\%$ $= \frac{1099}{1115} * 100\%$ $= \mathbf{98\%}$	$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} * 100\%$ $= \frac{965+113}{965+113+37+0} * 100\%$ $= \frac{1078}{1115} * 100\%$ $= \mathbf{96\%}$
$\text{Recall} = \frac{TP}{TP+FN} * 100\%$ $= \frac{961}{961+9} * 100\%$ $= \mathbf{99\%}$	$\text{Recall} = \frac{TP}{TP+FN} * 100\%$ $= \frac{965}{965+0} * 100\%$ $= \mathbf{100\%}$
$\text{Precision} = \frac{TP}{TP+FP} * 100\%$ $= \frac{961}{961+7} * 100\%$	$\text{Precision} = \frac{TP}{TP+FP} * 100\%$ $= \frac{965}{965+37} * 100\%$

Metode	
Naïve Bayes	Multinomial Naïve Bayes
= <b>99%</b>	= <b>96%</b>
$F1-Score = \frac{2*recall*precision}{recall+precision} * 100\%$ $= \frac{2*99*99}{99+99} * 100\%$ $= \mathbf{99\%}$	$F1-Score = \frac{2*recall*precision}{recall+precision} * 100\%$ $= \frac{2*100*96}{100+96} * 100\%$ $= \mathbf{97\%}$

#### 4.2 Analisis Hasil Evaluasi

Analisis hasil evaluasi merupakan tahap penting dalam penelitian ini, di mana fokusnya adalah membandingkan kinerja metode *Naïve Bayes* dan metode *Multinomial Naïve Bayes* dalam tugas klasifikasi data. Fokus utama dari analisis ini adalah untuk memahami sejauh mana kedua metode ini efektif dalam menangani situasi yang beragam yang terkait dengan data tersebut. Hasil dari analisis ini akan memberikan gambaran yang jelas tentang kinerja dari kedua metode dalam konteks tugas klasifikasi. Perbandingan data dan hasil akurasi yang didapat dari *Naïve Bayes* dan *Multinomial Naïve Bayes* dapat dilihat pada tabel berikut.

Tabel 4.7 Perbandingan model *naïve bayes* dan *multinomial naïve bayes*

	Naïve Bayes	Multinomial Naïve Bayes
Dataset	<ul style="list-style-type: none"> <li>• <i>WhatsApp</i> : 500</li> <li>• <i>Email</i> : 5736</li> <li>• <i>SMS</i> : 5565</li> </ul>	<ul style="list-style-type: none"> <li>• <i>WhatsApp</i> : 500</li> <li>• <i>Email</i> : 5736</li> <li>• <i>SMS</i> : 5565</li> </ul>
Akurasi	<ul style="list-style-type: none"> <li>• <i>WhatsApp</i> : 89%</li> <li>• <i>Email</i> : 99%</li> <li>• <i>SMS</i> : 98%</li> </ul>	<ul style="list-style-type: none"> <li>• <i>WhatsApp</i> : 81%</li> <li>• <i>Email</i> : 87%</li> <li>• <i>SMS</i> : 96%</li> </ul>

*Naïve Bayes* adalah algoritma pembelajaran mesin yang umum digunakan untuk klasifikasi dataset. Perbedaan antara *Naïve Bayes* dan *Multinomial Naïve Bayes* terletak pada asumsi distribusi data. *Naïve Bayes* mengasumsi bahwa data berdistribusi normal, sementara *Multinomial Naïve Bayes* mengasumsikan bahwa

data berdistribusi *multinomial*, artinya data terdiri dari angka-angka bulat yang merepresentasikan jumlah kemunculan setiap fitur.

Pada kasus teks, *Multinomial Naïve Bayes* digunakan karena teks dapat direpresentasikan sebagai vektor dengan hitungan kata atau frekuensi kata (*term frequency*). Namun, terkadang akurasi *Multinomial Naïve Bayes* bisa lebih rendah daripada *Naïve Bayes*. Hal ini dikarenakan *Multinomial Naïve Bayes* memiliki asumsi distribusi yang mungkin tidak cocok dengan data sebenarnya. Jumlah fitur yang besar dan sedikit contoh dapat menyebabkan kekurangan informasi, sementara jumlah fitur yang sangat besar dibandingkan contoh bisa menyebabkan *overfitting*. *Pre-processing* yang tidak tepat dan ketidakseimbangan kelas juga bisa mempengaruhi akurasi. Solusi untuk mengatasi nilai akurasi yang rendah adalah dengan melibatkan penyesuaian metode yang digunakan, termasuk penerapan tahapan *pre-processing* yang sesuai dengan metode tersebut. Tahapan *pre-processing* merupakan langkah penting dalam persiapan data sebelum dilakukan analisis atau klasifikasi. Dengan mempertimbangkan karakteristik khusus dari data yang dihadapi dapat meningkatkan kinerja model klasifikasi.

Perbandingan hasil dari kinerja model antara *Naïve Bayes* dan *Multinomial Naïve Bayes* tergantung pada jenis data dan tugas klasifikasi yang dihadapi. *Naïve Bayes* adalah pendekatan yang lebih umum digunakan dan cocok untuk data dengan atribut diskrit atau *continue*, sedangkan *Multinomial Naïve Bayes* lebih efektif dalam tugas klasifikasi teks dan data yang menggambarkan distribusi frekuensi atribut, seperti analisis sentimen atau klasifikasi dokumen.

Berikut grafik akurasi dari metode *Naïve Bayes* dan *Multinomial Naïve Bayes* yang disajikan dalam bentuk tabel berdasarkan kinerja klasifikasi yang dapat dilihat pada tabel di bawah ini:

Tabel 4.8 Grafik Perbandingan Akurasi Metode *Naïve Bayes* dan *Multinomial Naïve Bayes*

Dataset	Metode	
	Naïve Bayes	Multinomial Naïve Bayes
<i>WhatsApp</i>	<p>Akurasi Model Naïve Bayes</p>	<p>Akurasi Model Multinomial Naïve Bayes</p>
<i>SMS</i>	<p>Akurasi Model Naïve Bayes</p>	<p>Akurasi Model Multinomial Naïve Bayes</p>
<i>Email</i>	<p>Akurasi Model Naïve Bayes</p>	<p>Akurasi Model Multinomial Naïve Bayes</p>

Dari hasil perbandingan ini, dapat disimpulkan bahwa kedua metode memiliki keunggulan dan kelemahan masing-masing. Meskipun metode *Naïve Bayes* lebih unggul dalam beberapa aspek, metode *Multinomial Naïve Bayes* tetap menunjukkan kinerja yang baik. Tingkat akurasi juga dapat dipengaruhi oleh jumlah data yang digunakan. Semakin banyak jumlah data yang digunakan maka nilai akurasi yang dihasilkan akan semakin tinggi. Pemahaman ini memungkinkan peneliti dan praktisi untuk membuat keputusan yang lebih baik dalam memilih metode yang paling sesuai, dengan mempertimbangkan data yang akan dihadapi.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil pembahasan dan penelitian yang telah diuraikan pada bab sebelumnya, maka dapat disimpulkan sebagai berikut:

1. Penelitian ini menggunakan tiga jenis dataset yaitu *WhatsApp*, *Email* dan *SMS*. Nilai akurasi metode *Naïve Bayes* menggunakan dataset *WhatsApp* sebesar 89%, dengan menggunakan dataset *Email* nilai akurasinya mencapai 99% dan dataset *SMS* berhasil mencapai nilai akurasi 98%. Sementara itu nilai akurasi dari metode *Multinomial Naïve Bayes* menggunakan dataset *WhatsApp* adalah 81%, dengan implementasi dataset *Email* memperoleh nilai akuarasi sebesar 87%, dari dataset *SMS* memperoleh akurasi sebesar 96%. Nilai akurasi ini dapat dikategorikan baik.
2. Perbandingan rata-rata akurasi antara metode *Naïve Bayes* dan *Multinomial Naïve Bayes*, metode *Naïve Bayes* memiliki akurasi yang lebih tinggi dengan rata-rata 95%, sementara metode *Multinomial Naïve Bayes* sebesar 88%. Metode *Naïve Bayes* lebih unggul karena fleksibilitasnya dalam menangani berbagai jenis data tanpa asumsi spesifikasi tentang distribusi probabilitas, sehingga memungkinkan untuk lebih baik dalam variasi dataset yang berbeda.

#### 5.2 Saran

Berdasarkan kesimpulan dari hasil dan pembahasan pada penelitian ini, maka penulis mengemukakan beberapa saran yang dapat digunakan untuk pengembangan lanjutan dari penelitian ini, sebagai berikut:

1. Sistem klasifikasi *message* ini bisa dikembangkan dengan menggunakan metode klasifikasi lain selain dari metode *Naïve Bayes* dan *Multinomial Naïve Bayes* untuk mengetahui perbedaan tingkat akurasinya.
2. Melakukan pembangunan model *Naïve Bayes* dan *Multinomial Naïve Bayes* dengan arsitektur dan parameter yang berbeda.

## DAFTAR PUSTAKA

- Aripin, J. J. (2019). *Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi pada BPR Pantura*. <https://repository.nusamandiri.ac.id/index.php/repo/viewitem/13890>
- Bahari, M. F. (2022). Analisa Dan Implementasi Keamanan Pesan Chatting Menggunakan Algoritma Challenge Response. *JUSSI: Jurnal Sains Dan Teknologi Informasi*, 1(2), 49–53.
- Dewi, F. K. S. (2021). Klasifikasi Berita Menggunakan Metode Multinomial Naïve Bayes. *SCAN - Jurnal Teknologi Informasi Dan Komunikasi*, 16(3), 1–8. <https://doi.org/10.33005/scan.v16i3.2870>
- Fitria, A., & Azis, H. (2018). Analisis Kinerja Sistem Klasifikasi Skripsi menggunakan Metode Naïve Bayes Classifier. *Prosiding Seminar Nasional Ilmu Komputer Dan Teknologi Informasi*, 3(2), 102–106.
- Gumilar, M. D., Sembiring, F., Erfina, A., Studi, P., Informasi, S., Putra, U. N., Method, D., Architecture, T., & Method, D. (n.d.). *IMPLEMENTASI PROGRESSIVE WEB APP PADA SISTEM INFORMASI E-LEARNING UNTUK*.
- Haryono, H., Palupiningsih, P., Asri, Y., & Handayani, A. N. S. (2018). Klasifikasi Pesan Gangguan Pelanggan Menggunakan Metode Naive Bayes Classifier. *Kilat*, 7(2), 100–108. <https://doi.org/10.33322/kilat.v7i2.354>
- Hidayat, A. (2023). *KLASIFIKASI SPAM EMAIL MENGGUNAKAN*. 3(2), 1–14.
- Ihsan, C. N. (2021). Klasifikasi Data Radar Menggunakan Algoritma Convolutional Neural Network (CNN). *DoubleClick: Journal of Computer and Information Technology*, 4(2), 115. <https://doi.org/10.25273/doubleclick.v4i2.8188>
- Juli, V. N., Ollivia, A., & Pratiwi, C. (2023). *Klasifikasi Jenis Anggur Berdasarkan Bentuk Daun Menggunakan Convolutional Neural Network Dan K-Nearest Neighbor*. 3(2).
- Migunani, & Kevin Aditama. (2020). Pemanfaatan Natural Language Processing Dan Pattern Matching Dalam Pembelajaran Melalui Guru Virtual. *Elkom : Jurnal Elektronika Dan Komputer*, 13(1), 121–133. <https://doi.org/10.51903/elkom.v13i1.187>
- Mustofa, H., & Mahfudh, A. A. (2019). Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes. *Walisongo Journal of Information Technology*, 1(1), 1. <https://doi.org/10.21580/wjit.2019.1.1.3915>
- Mutawalli, L., Zaen, M. T. A., & Bagye, W. (2019). KLASIFIKASI TEKS SOSIAL MEDIA TWITTER MENGGUNAKAN SUPPORT VECTOR MACHINE (Studi Kasus Penusukan Wiranto). *Jurnal Informatika Dan Rekayasa Elektronik*, 2(2), 43. <https://doi.org/10.36595/jire.v2i2.117>

- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 5(2), 697–711.
- Plianda, I. A., & Indrayani, R. (2022). Analisa dan Perbandingan Performa Tools Forensik Digital pada Smartphone Android menggunakan Instant Messaging Whatsapp. *Jurnal Media Informatika Budidarma*, 6(1), 500. <https://doi.org/10.30865/mib.v6i1.3487>
- Putro, H. F., Vlandari, R. T., & Saptomo, W. L. Y. (2020). Penerapan Metode Naive Bayes Untuk Klasifikasi Pelanggan. *Jurnal Teknologi Informasi Dan Komunikasi (TIKomsin)*, 8(2). <https://doi.org/10.30646/tikomsin.v8i2.500>
- Saluky, S. (2018). Tinjauan Artificial Intelligence untuk Smart Government. *ITEJ (Information Technology Engineering Journals)*, 3(1), 8–16. <https://doi.org/10.24235/itej.v3i1.22>
- Soen, G. I. E., Marlina, & Renny. (2022). Implementasi Cloud Computing dengan Google Colaboratory Pada Aplikasi Pengolah Data Zoom Participants. *Journal Informatic Technology And Communication*, 6(1), 24–30.
- Sriyano, C. S., & Setiawan, E. B. (2021). Pendeteksian Berita Hoax Menggunakan Naive Bayes Multinomial Pada Twitter dengan Fitur Pembobotan TF-IDF. *E-Proceeding of Engineering*, 8(2), 3396–3405.
- Yunanto, R., Purfini, A. P., & Prabuwisesa, A. (2021). Survei Literatur: Deteksi Berita Palsu Menggunakan Pendekatan Deep Learning. *Jurnal Manajemen Informatika (JAMIKA)*, 11(2), 118–130. <https://doi.org/10.34010/jamika.v11i2.536>



## LAMPIRAN

Adapun lampiran lengkap hasil dari penelitian ini sebagai berikut :

### **Skenario 1. Source Code metode Naïve Bayes menggunakan dataset (WhatsApp/Email/SMS)**

Code berikut merupakan program *python* untuk mengimport *library* dan modul yang dibutuhkan.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import string
import re
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
from nltk.corpus import stopwords

!pip install tweet-preprocessor
!pip install matplotlib
!pip install transformers
!pip install emoji
!pip install sastrawi
!pip install pySastrawi
!pip install vaderSentiment

!pip install Sastrawi
!pip install swifter
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

import string
import pandas as pd
import numpy as np
import nltk
nltk.download('punkt')
import re
from nltk.corpus import stopwords
```

Code dibawah ini digunakan untuk mengimport dataset dan menampilkan dataset (WhatsApp/Email/SMS).

```
# Import pandas
```

```
import pandas as pd

# Membaca dataset
df = pd.read_csv('/content/WhatsApp (4).csv', encoding = 'ISO-8859-1', error_bad_lines=False)
# menampilkan data
df.head()
```

Code ini digunakan untuk mengganti nama kolom secara otomatis.

```
# menamai ulang kolom dan menghapus kolom yang tidak diperlukan
df.rename(columns={'v1': 'Label', 'v2': 'text'}, inplace=True)
df.drop(df.columns[2:5], axis=1, inplace=True)
# menampilkan data yang telah di modif
df.head()
```

Code dibawah ini digunakan untuk melakukan *pro-processing* data dengan tahapan *case folding, tokenized, stopword removal, stemming*.

```
import nltk
nltk.download('stopwords')
nltk.download('stopwords-id')

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer
import re
ps = PorterStemmer()

messages = []

for i in range(0, df.shape[0]):
    # Memeriksa apakah nilai dalam kolom 'text' adalah string
    if isinstance(df['teks'][i], str):
        # Jika ya, lanjutkan dengan pemrosesan
        review = re.sub('[^a-zA-Z]', ' ', df['teks'][i])
        review = review.lower()
        review = review.split()
        review = [ps.stem(word) for word in review if not word
in stopwords.words('indonesian')]
        review = ' '.join(review)
        messages.append(review)
    else:
        # Jika bukan string, tambahkan string kosong atau
tindakan lain sesuai kebutuhan
        messages.append('')
# Replace the 'teks' column with tokenized message
df['teks'] = messages
```

```
# Display the first five rows
display(df.head())
```

Code berikut akan membagi dataset yang telah melalui tahap *pre-processing* menjadi data *training* dan data *testing* dengan perbandingan 80:20, lalu menampilkan jumlah dari masing-masing data *training* dan data *testing*.

```
# Menentukan features dan labels
x = df['teks']
y = df['label'].tolist()

# Menentukan jumlah data latih dan data uji
train_size = 0.8
num_training = int(train_size * len(df))
num_test = len(df) - num_training
# Menampilkan jumlah data latih dan data uji
print(f>Data-training: {num_training}")
print(f>Data-testing: {num_test}")

# Memisahkan data latih dan data uji menggunakan indeks
slicing
x_train, y_train = x[:num_training], y[:num_training]
x_test, y_test = x[num_training:], y[num_training:]

# Output jumlah data latih dan data uji
print(f>Jumlah data train: {len(x_train)}")
print(f>Jumlah data test: {len(x_test)}")
```

Code ini digunakan untuk melatih program berdasarkan data yang sudah kita simpan.

```
# Fungsi untuk membuat kumpulan data spam dan non-spam
def training(x, y):
    d_non_spam, d_spam = {}, {}
    for i in range(len(x)):
        if isinstance(x[i], str): # Periksa apakah x[i]
            adalah string
            sentence = x[i].split()
            for word in sentence:
                if y[i] == 0:
                    if word in d_non_spam:
                        d_non_spam[word] += 1
                    else:
                        d_non_spam[word] = 1
                else:
                    if word in d_spam:
                        d_spam[word] += 1
                    else:
```

```

        d_spam[word] = 1
    return d_non_spam, d_spam

# Filter hanya data yang merupakan string dalam x_train
x_train_filtered = [item for item in x_train if
                    isinstance(item, str)]

# Panggil fungsi training dengan data yang telah difilter
non_spam_counter, spam_counter = training(x_train_filtered,
                                          y_train)

# Menghitung jumlah data spam dan non spam keseluruhan
total_non_spam, total_spam = sum(non_spam_counter.values()),
sum(spam_counter.values())

```

*Code ini untuk menghitung probabilitas bahwa suatu pesan adalah non-spam atau spam.*

```

# menghitung probabilitas non spam dan spam
prob_non_spam = y.count(0) / len(y)
prob_spam = y.count(1) / len(y)

prob_non_spam + prob_spam

```

*Code ini untuk membuat model klasifikasi terhadap data training.*

```

# Fungsi untuk melakukan klasifikasi terhadap data uji
def classify(unknown):
    prob_spam_as_non_spam, prob_spam_as_spam = 1, 1

    # Inputan dipisah per katanya
    review_words = unknown.split()

    # Perulangan untuk mendapatkan probabilitas SMS sebagai
    non spam dan spam
    for word in review_words: # Mengganti review_word menjadi
review_words
        word_in_non_spam = non_spam_counter.get(word, 0)
        try:
            prob_spam_as_non_spam *= (word_in_non_spam + 1) /
(total_non_spam + len(non_spam_counter))
        except ZeroDivisionError:
            prob_spam_as_non_spam = 0
        word_in_spam = spam_counter.get(word, 0)
        prob_spam_as_spam *= (word_in_spam + 1) / (total_spam
+ len(spam_counter))

```

```

    # Menghitung probabilitas hasil akhir untuk menentukan
    hasil prediksi
    final_non_spam, final_spam = prob_spam_as_non_spam *
    prob_non_spam, prob_spam_as_spam * prob_spam

    # Pengondisian hasil prediksi
    if final_non_spam > final_spam:
        return 0
    else:
        return 1

# Fungsi untuk menghitung evaluasi matriks kinerja
pengklasifikasian
def find_evaluation_metrics(validation_set,
validation_labels):
    tp, tn, fp, fn = 0, 0, 0, 0

    for i in range(len(validation_set)): # Mengganti "for 1
in" menjadi "for i in"
        guess = classify(validation_set.iat[i])
        if guess == 1 and validation_labels[i] == 1:
            tp += 1
        elif guess == 0 and validation_labels[i] == 0: #
Mengganti "validation_labels[1]" menjadi
"validation_labels[i]"
            tn += 1
        elif guess == 1 and validation_labels[i] == 0: #
Mengganti "validation_labels[1]" menjadi
"validation_labels[i]"
            fp += 1
        else:
            fn += 1
    accuracy = (tp + tn) / len(validation_set)
    recall = tp / (tp + fn)

    # handle perhitungan yang hasilnya 0
    try:
        precision = tp / (tp + fp)
    except ZeroDivisionError:
        precision = 0

    try:
        f_1 = 2 * precision * recall / (precision + recall)
    except ZeroDivisionError:
        f_1 = 0

    return accuracy, recall, precision, f_1

```

```

# Memanggil fungsi find_evaluation_metrics dengan kolom 'teks'
dari x_test dan kolom 'label' dari y_test
accuracy, recall, precision, f_1 =
find_evaluation_metrics(x_test, y_test)

# Menampilkan evaluasi matriks kinerja pengklasifikasian
print(f"Akurasi: {accuracy:.3f}")
print(f"Recall: {recall:.3f}")
print(f"Precision: {precision:.3f}")
print(f"F1-Score: {f_1:.3f}")

```

Code ini digunakan untuk mengonversi teks kedalam dataset .

```

from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd
# Konversi Series ke list dan mengganti nilai np.nan dengan
string kosong
x_train_list = [str(x) if pd.notna(x) else '' for x in
x_train]
x_test_list = [str(x) if pd.notna(x) else '' for x in x_test]

# Inisialisasi objek CountVectorizer
counter = CountVectorizer()

# Mengaplikasikan pada data training dan data uji
training_counts = counter.fit_transform(x_train_list)
test_counts = counter.transform(x_test_list)

# membuat objek pengklasifikasian Naive Bayes yaitu
MultinomialNB
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()

# kemudian melatih objek pengklasifikasi menggunakan data
latih sesuai format objek CountVectorizer dengan labelnya
clf.fit(training_counts, y_train)
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

```

Code ini untuk membuat model klasifikasi dengan data *testing*.

```

# mengklasifikasikan data testing
# menebak data testing
pred_y = clf.predict(test_counts)

```

Code ini digunakan untuk menghitung akurasi, *recall*, *precision*, dan *F1-Score*.

```

# utk 4 matrix
# Melakukan prediksi pada data uji
pred_y = clf.predict(test_counts)
# Menghitung konfusi matriks

```

```

conf_matrix = confusion_matrix(y_test, pred_y)

# Menampilkan matriks kebingungan
print("Confusion Matrix:")
print(conf_matrix)

# Menampilkan nilai TP, TN, FP, FN
tp = conf_matrix[0, 0]
tn = conf_matrix[1, 1]
fp = conf_matrix[1, 0]
fn = conf_matrix[0, 1]

# Menampilkan nilai TP, TN, FP, FN
print(f"\nTrue Positive (TP): {tp}")
print(f"True Negative (TN): {tn}")
print(f"False Positive (FP): {fp}")
print(f"False Negative (FN): {fn}")

# menghitung akurasi
accuracy = (tp + tn) / len(x_test)
recall = tp / (tp + fn)
precision = tp / (tp + fp)
f_1 = 2 * precision * recall / (precision + recall)
print("\nAkurasi: ", accuracy)
print("recall: ", recall)
print("precision: ", precision)
print("f_1: ", f_1)

# utk 9 matrix
# Melakukan prediksi pada data uji
pred_y = clf.predict(test_counts)

# Menghitung konfusi matriks
conf_matrix = confusion_matrix(y_test, pred_y)

# Menampilkan matriks kebingungan
print("Confusion Matrix:")
print(conf_matrix)

# Menampilkan nilai TP, TN, FP, FN
tp = conf_matrix[0, 0]
tn = conf_matrix[1, 1] + conf_matrix[1,2] + conf_matrix[2,1] +
conf_matrix[2,2]
fp = conf_matrix[1, 0] + conf_matrix[2,0]
fn = conf_matrix[0, 1] + conf_matrix[0,2]
# Menampilkan nilai TP, TN, FP, FN

```

```

print(f"\nTrue Positive (TP): {tp}")
print(f"True Negative (TN): {tn}")
print(f"False Positive (FP): {fp}")
print(f"False Negative (FN): {fn}")

# menghitung akurasi
accuracy = (tp + tn) / len(x_test)
recall = tp / (tp + fn)
precision = tp / (tp + fp)
f_1 = 2 * precision * recall / (precision + recall)
print("\nAkurasi: ", accuracy)
print("recall: ", recall)
print("precision: ", precision)
print("f_1: ", f_1)

```

Code ini untuk menghitung dan menampilkan diagram *confusion matrix* dalam bentuk heatmap menggunakan **library seaborn** dan **matplotlib.pyplot** serta menampilkan *true label* dan prediksi label.

```

from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
# Melakukan prediksi pada data uji
pred_y = clf.predict(test_counts)

# Menghitung confusion matrix
conf_matrix = confusion_matrix(y_test, pred_y)

# Membuat heatmap untuk menampilkan confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=['non-spam', 'spam'], yticklabels=['non-spam',
'spam'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

```

Code ini berfungsi untuk menghitung dan mencetak nilai akurasi, *recall*, *precision* dan *f1-score*. performa model klasifikasi ini akan memberikan informasi lebih rinci daripada hanya menggunakan *confusion matrix*.

```

from sklearn.metrics import accuracy_score, recall_score,
precision_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
import pandas as pd

```



```

#(your previous code for loading data and other preprocessing)
# Inisialisasi objek pengklasifikasi Naive Bayes
clf = MultinomialNB()

# Konversi Series ke list dan mengganti nilai np.nan dengan
string kosong
x_list = [str(x) if pd.notna(x) else '' for x in x]

# Inisialisasi variabel untuk menyimpan hasil terbaik
best_accuracy = 0.0
best_split = 0

# Varying train/test split from 10% to 90%
for split in range(10, 100, 10):
    # Membagi data menjadi data latih dan data uji
    x_train, x_test, y_train, y_test =
train_test_split(x_list, y, test_size=split/100,
random_state=42)

    # Inisialisasi objek CountVectorizer
    counter = CountVectorizer()

    # Mengaplikasikan pada data training dan data uji
    training_counts = counter.fit_transform(x_train)
    test_counts = counter.transform(x_test)

    # Melatih model
    clf.fit(training_counts, y_train)

    # Melakukan prediksi pada data uji
    pred_y = clf.predict(test_counts)

    # Menghitung dan menyimpan akurasi
    accuracy = accuracy_score(y_test, pred_y)
    if accuracy > best_accuracy:
        best_accuracy = accuracy
        best_split = split

# Print the evaluation metrics for the best split
print(f"Best Split: {best_split}%")
print(f"Best Accuracy: {best_accuracy:.3f}")

# Matriks kinerja pengklasifikasi untuk data uji terbaik
x_train, x_test, y_train, y_test = train_test_split(x_list, y,
test_size=best_split/100, random_state=42)
training_counts = counter.fit_transform(x_train)
test_counts = counter.transform(x_test)

```

```

clf.fit(training_counts, y_train)
pred_y = clf.predict(test_counts)

# Menampilkan matriks kinerja pengklasifikasi
print(f"Akurasi: {accuracy_score(y_test, pred_y):.3f}")
print(f"Recall (berbobot): {recall_score(y_test, pred_y,
average='weighted'):.3f}")
print(f"Presisi: {precision_score(y_test, pred_y,
average='weighted'):.3f}")
print(f"Skor F1: {f1_score(y_test, pred_y,
average='weighted'):.3f}")

```

Code ini digunakan untuk memprediksi data apakah *spam* atau *non-spam* menurut model.

```

# Kode untuk mengonversi label menjadi spam dan non-spam
def spam_or_non_spam(x):
    return 'non-spam' if x == 0 else 'spam'

# Untuk mengulang label prediksi
for review, label in zip(x_test, pred_y):
    print(f"\n{review}\n --> {spam_or_non_spam(label)}")

```

## **Skenario 2. Source Code metode Multinomial Naïve Bayes menggunakan dataset (WhatsApp/Email/SMS)**

Code berikut merupakan program *python* untuk mengimport *library* dan modul yang dibutuhkan.

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import string
import re
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from nltk.tokenize import word_tokenize
from sklearn.linear_model import LogisticRegression
from nltk.probability import FreqDist
from nltk.corpus import stopwords

!pip install tweet-preprocessor
!pip install matplotlib
!pip install transformers
!pip install emoji
!pip install sastrawi

```

```

!pip install pySastrawi
!pip install vaderSentiment

!pip install Sastrawi
!pip install swifter
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

import string
import pandas as pd
import numpy as np
import nltk
nltk.download('punkt')
import re
from nltk.corpus import stopwords

```

Code dibawah ini digunakan untuk mengimport dataset dan menampilkan dataset (*WhatsApp/Email/SMS*).

```

# upload dataset spam teks
from google.colab import files
uploaded = files.upload()

# mengubah dataset menjadi dataframe dengan variabel df
import pandas as pd
df = pd.read_csv('WhatsApp (4).csv', encoding='ISO-8859-1',
error_bad_lines=False)
# menampilkan 5 sampel teratas pada dataframe
df.head()

```

Code ini untuk menampilkan rangkuman statistik.

```
display(df.describe())
```

Code ini digunakan untuk menampilkan informasi ringkas tentang *dataframe*.

```

# print the info
print(df.info())

```

Code digunakan untuk menghitung dan menampilkan jumlah data *spam* dan *non-spam*.

```

# melihat category/label dalam dataset spam message
df['label'].value_counts()

```

Code ini untuk membuat dan menampilkan grafik yang menunjukkan jumlah kategori dari 'label' pada *dataframe*.

```

import seaborn as sns
import matplotlib.pyplot as plt
# Anda harus memiliki DataFrame 'df' dengan kolom 'label'

```

```
sns.countplot(data=df, x='label')
plt.title("Category Counts")
plt.show()
```

*Code ini digunakan untuk mengubah nilai-nilai kategori dalam kolom 'label' pada dataframe.*

```
from sklearn.preprocessing import LabelEncoder
# Initialize the Label Encoder
le = LabelEncoder()

# Encode the categories
df['Label_Encoding'] = le.fit_transform(df['label'])
# display the first five rows again to see the result
display(df.head())
```

*Code dibawah ini digunakan untuk melakukan pro-processing data dengan tahapan case folding, tokenized, stopwords removal, stemming.*

```
import nltk
nltk.download('stopwords')
nltk.download('stopwords-id')

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer
import re
ps = PorterStemmer()

messages = []

for i in range(0, df.shape[0]):
    # Memeriksa apakah nilai dalam kolom 'text' adalah string
    if isinstance(df['teks'][i], str):
        # Jika ya, lanjutkan dengan pemrosesan
        review = re.sub('[^a-zA-Z]', ' ', df['teks'][i])
        review = review.lower()
        review = review.split()
        review = [ps.stem(word) for word in review if not word
in stopwords.words('indonesian')]
        review = ' '.join(review)
        messages.append(review)
    else:
        # Jika bukan string, tambahkan string kosong atau
tindakan lain sesuai kebutuhan
        messages.append('')

# Replace the 'teks' column with tokenized message
```

```
df['teks'] = messages
```

```
# Display the first five rows  
display(df.head())
```

Code ini digunakan untuk mengubah nilai-nilai dalam kolom 'label' dari *dataframe* menjadi representasi numerik menggunakan teknik yang disebut 'label encoding'.

```
# Initialize the Label Encoder  
le = LabelEncoder()  
  
# Encode the categories and create 'Label_Encoding' column  
df['Label_Encoding'] = le.fit_transform(df['label'])  
# merubah variabel Message=x dan Label_Encoding=y  
x = df['teks']  
y = df['Label_Encoding']  
  
# membagi data menjadi data latih (train) dan data uji (test)  
from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test = train_test_split(x, y,  
test_size=0.2, random_state=42)
```

Code ini digunakan untuk menggantikan nilai kosong dalam data latih dan data uji dengan string kosong dan menghitung representasi TF-IDF dari teks pada data latih dan data uji menggunakan *TfidfVectorizer*.

```
from sklearn.feature_extraction.text import TfidfVectorizer  
# Mengganti nilai np.nan dengan string kosong pada x_train dan  
x_test  
x_train = x_train.fillna('')  
x_test = x_test.fillna('')  
  
# Selanjutnya, Anda dapat membuat vectorizer dan menghitung  
TF-IDF seperti yang telah Anda lakukan  
vectorizer = TfidfVectorizer(strip_accents='ascii')  
tfidf_train = vectorizer.fit_transform(x_train)  
tfidf_test = vectorizer.transform(x_test)
```

Code ini digunakan untuk membuat, melatih dan menguji model klasifikasi *Multinomial Naive Bayes* pada data teks yang telah diubah menjadi representasi TF-IDF.

```
# membuat model  
from sklearn.naive_bayes import MultinomialNB  
  
# Multinomial Naive Bayes Classifier  
nb=MultinomialNB()  
  
# fit model
```

```
nb.fit(tfidf_train, y_train)
print("Accuracy:", nb.score(tfidf_test, y_test))
```

Code ini untuk menghitung dan menampilkan diagram *confusion matrix* dalam bentuk heatmap menggunakan **library seaborn** dan **matplotlib.pyplot** serta menampilkan *true label* dan prediksi label.

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix
# Melakukan prediksi pada data uji
y_pred = nb.predict(tfidf_test)

# Menghitung confusion matrix
confusion = confusion_matrix(y_test, y_pred)

# Membuat tampilan confusion matrix dengan seaborn
plt.figure(figsize=(8, 6))
sns.heatmap(confusion, annot=True, fmt='d', cmap='Blues',
            xticklabels=['non-spam', 'spam'], yticklabels=['non-spam',
            'spam'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

Code ini berfungsi untuk menghitung dan mencetak nilai akurasi, *recall*, *precision* dan *f1-score*. performa model klasifikasi ini akan memberikan informasi lebih rinci daripada hanya menggunakan *confusion matrix*.

```
# utk 4 matrix
# print the labels
y_pred = nb.predict(tfidf_test)

# Menghitung konfusi matriks
conf_matrix = confusion_matrix(y_test, y_pred)

# Menampilkan matriks kebingungan
print("Confusion Matrix:")
print(conf_matrix)

# Menampilkan nilai TP, TN, FP, FN
tp = conf_matrix[0, 0]
tn = conf_matrix[1, 1]
fp = conf_matrix[1, 0]
fn = conf_matrix[0, 1]

# Menampilkan nilai TP, TN, FP, FN
```

```

print(f"\nTrue Positive (TP): {tp}")
print(f"True Negative (TN): {tn}")
print(f"False Positive (FP): {fp}")
print(f"False Negative (FN): {fn}")

# menghitung akurasi
accuracy = (tp + tn) / len(x_test)
recall = tp / (tp + fn)
precision = tp / (tp + fp)
f_1 = 2 * precision * recall / (precision + recall)
print("\nAkurasi: ", accuracy)
print("recall: ", recall)
print("precision: ", precision)
print("f_1: ", f_1)

# utk 9 matrix
# print the labels
y_pred = nb.predict(tfidf_test)

# Menghitung konfusi matriks
conf_matrix = confusion_matrix(y_test, y_pred)

# Menampilkan matriks kebingungan
print("Confusion Matrix:")
print(conf_matrix)

# Menampilkan nilai TP, TN, FP, FN
tp = conf_matrix[0, 0]
tn = conf_matrix[1, 1] + conf_matrix[1,2] + conf_matrix[2,1] +
conf_matrix[2,2]
fp = conf_matrix[1, 0] + conf_matrix[2,0]
fn = conf_matrix[0, 1] + conf_matrix[0,2]

# Menampilkan nilai TP, TN, FP, FN
print(f"\nTrue Positive (TP): {tp}")
print(f"True Negative (TN): {tn}")
print(f"False Positive (FP): {fp}")
print(f"False Negative (FN): {fn}")

# menghitung akurasi
accuracy = (tp + tn) / len(x_test)
recall = tp / (tp + fn)
precision = tp / (tp + fp)
f_1 = 2 * precision * recall / (precision + recall)
print("\nAkurasi: ", accuracy)
print("recall: ", recall)

```

```
print("precision: ", precision)
print("f_1: ", f_1)
```

Code ini digunakan untuk memprediksi data apakah *spam* atau *non-spam* menurut model.

```
# Define a function to convert label encoding to "spam" or
"non-spam"
def spam_or_non_spam(x):
    return 'non-spam' if x == 0 else 'spam'

# Iterate through the predicted labels and the corresponding
text data
for review, label in zip(x_test, y_pred):
    print(f"\n{review}\n --> {spam_or_non_spam(label)}")
```

