

**PERBANDINGAN METODE SUPPORT VECTOR MACHINE
DAN NAIVE BAYES TERHADAP PENGGUNAAN
ARTIFICIAL INTELLIGENCE DALAM PEMBUATAN
SKRIPSI PADA MEDIA SOSIAL X**

TUGAS AKHIR

**Diajukan Oleh :
Albiran Nisa
NIM. 200705002**

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



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2024 M / 1446 H**

LEMBAR PERSETUJUAN

PERBANDINGAN METODE SUPPORT VECTOR MACHINE DAN NAIVE BAYES TERHADAP PENGGUNAAN ARTIFICIAL INTELLIGENCE DALAM PEMBUATAN SKRIPSI PADA MEDIA SOSIAL X

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
dalam Ilmu/Prodi Teknologi Informasi

Oleh:
ALBIRAN NISA
NIM. 200705002
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I



Malahayati, M.T
NIP. 198301272015032003

Pembimbing II



Hendri Ahmadian, M.I.M
NIP. 198301042014031002

Mengetahui,
Ketua Program Studi Teknologi Informasi



Malahayati, M.T
NIP. 198301272015032003

LEMBAR PENGESAHAN

PERBANDINGAN METODE SUPPORT VECTOR MACHINE DAN NAIVE BAYES TERHADAP PENGGUNAAN ARTIFICIAL INTELLIGENCE DALAM PEMBUATAN SKRIPSI PADA MEDIA SOSIAL X


TUGAS AKHIR

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

Pada Hari/Tanggal: Selasa/20 Agustus 2024
di Darussalam, Banda Aceh

Panitia ujian munaqasyah tugas akhir

Ketua


Malahayati, M.T
NIP. 198301272015032003


Sekretaris


Hendri Ahmadian, M.I.M
NIP. 198301042014031002

Penguji I

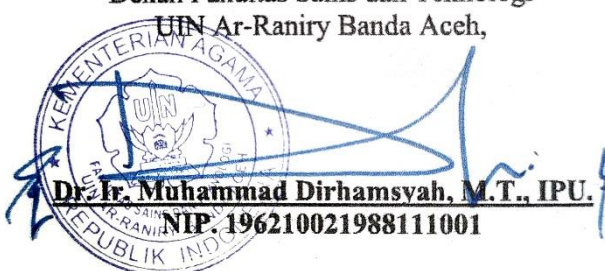

Khairan AR, M.Kom
NIP. 198607042014031001

Penguji II


Fathiah, M.Eng
NIP. 198606152019032010

Mengetahui:

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


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

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini:

Nama : Albiran Nisa
NIM : 200705002
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul : Perbandingan Metode *Support Vector Machine* dan *Naive Bayes* Terhadap Penggunaan *Artificial Intelligence* Dalam Pembuatan Skripsi Pada Media Sosial X

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 karya 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 bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai 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, 20 Agustus
2024

Yang Menyatakan



Albiran Nisa

ABSTRAK

Nama : Albiran Nisa
NIM : 200705002
Program Studi : Teknologi Informasi
Judul : Perbandingan Metode *Support Vector Machine* dan *Naive Bayes* Terhadap Penggunaan *Artificial Intelligence* Dalam Pembuatan Skripsi Pada Media Sosial X
Tanggal Sidang : 20 Agustus 2024
Jumlah Halaman : 82 Halaman
Pembimbing I : Malahayati, M.T
Pembimbing II : Hendri Ahmadian, M.I.M
Kata Kunci : *Super Vector Machine*, *Naive Bayes*, Analisis Sentimen, Media Sosial X

Kemajuan perkembangan teknologi saat ini memudahkan pekerjaan serta memenuhi kebutuhan masyarakat termasuk dalam bidang pendidikan khususnya mahasiswa. Salah satu teknologi tersebut adalah *Artificial Intelligence* (kecerdasan buatan) atau disingkat AI merupakan teknologi yang serupa dengan mesin yang dapat meniru perilaku manusia. Namun, seiring meningkatnya penggunaan aplikasi berbasis AI ini menimbulkan ketergantungan pada mahasiswa dengan *tools* yang telah disediakan. Sehingga, mengurangi kemampuan mereka untuk berpikir secara kritis serta mengalami kesulitan dalam memecahkan masalah secara mandiri. Tujuan dari penelitian ini adalah untuk menganalisis opini masyarakat media sosial x dengan membandingkan metode SVM dan *Naive Bayes*. Hasil yang diperoleh dari membandingkan kedua metode tersebut adalah, nilai akurasi SVM lebih besar dengan nilai akurasi sebesar 100% sedangkan nilai akurasi *Naive Bayes* sebesar 90%. Ini membuktikan bahwa SVM lebih akurat daripada *Naive Bayes*.

KATA PENGANTAR

Dengan menyebut nama Allah SWT yang maha pengasih lagi maha penyayang. Puji syukur penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Perbandingan Metode *Support Vector Machine* dan *Naive Bayes* Terhadap Penggunaan *Artificial Intelligence* Dalam Pembuatan Skripsi Pada Media Sosial X”. 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. Ayahanda Syaiful Azmi dan Ibunda Nurmala Sari, terimakasih untuk segala kasih dan sayang yang tidak pernah luntur untuk penulis. Terimakasih selalu memberikan dukungan dan doa yang tidak pernah berhenti untuk penulis sehingga penulis mampu bertahan dan menyelesaikan pendidikan sampai sarjana.
2. Nenek Siti Aisyah, terimakasih selalu menanyakan kabar dan memberikan dukungan serta doa untuk penulis.
3. Adinda Putri Syakila dan Rafiq Khairiri yang merupakan adik kandung dari penulis, terimakasih karena sudah ikut serta memberikan banyak dukungan, semangat, serta doa dan kasih sayang selama penyusunan tugas akhir ini.
4. Ibu Malahayati, M.T selaku pembimbing I dan Bapak Hendri Ahmadian, M.I.M selaku pembimbing II, sungguh penulis berterimakasih atas segala bimbingan, kritik, serta saran dan masukan untuk penulis selama pengerjaan tugas akhir ini. Terimakasih selalu meluangkan waktu untuk membimbing penulis. Semoga jerih payah ibu dan bapak terbayarkan dan dilimpahkan keberkahan serta kesehatan.

5. Ketua program studi Teknologi Informasi, Ibu Malahayati, M.T yang juga menjadi pembimbing penulis, terimakasih untuk segala bimbingan dan arahan dalam penyusunan tugas akhir penulis.
6. Ibu Cut Ida Rahmadiana, S.Si selaku staff prodi Teknologi Informasi yang sangat senantiasa membantu penulis dalam pemberkasan administrasi selama ini.
7. Bapak Dr. Ir, M. Dirhamsyah, M.T.,IPU selaku Dekan Fakultas Sains dan Teknologi Informasi dan selaku Penasehat Akademik penulis yang telah memberi dukungan, doa dan motivasi untuk penyusunan tugas akhir penulis.
8. Bapak dan Ibu dosen prodi Teknologi Informasi yang telah memberikan ilmu kepada penulis dalam bidang Teknologi Informasi.
9. Sahabat seperjuangan selama penyusunan tugas akhir penulis Alfiani, terimakasih sudah mengizinkan penulis untuk tinggal sementara bersama denganmu. Terimakasih karena sudah bersedia menghabiskan banyak waktu untuk hal-hal baik dan sama-sama meyakinkan bahwa kita akan selesai pada perkuliahan ini tepat waktu.
10. Semua sahabat di masa sekolah menengah atas, terutama Dea Nanda Putri Purwani, Dwi Ranti, dan Siti Haliza, terimakasih masih menjalin hubungan baik dengan penulis sampai saat ini. Terimakasih selalu memberikan pencerahan, motivasi dan hal-hal baik walaupun hanya lewat perantaraan media sosial. Terimakasih selalu mendukung dan mendoakan segala hal baik untuk penulis disini. Semoga kalian selalu diberikan kemudahan, kesehatan dan keberkahan.
11. Semua teman-teman terdekat penulis dan seluruh teman prodi teknologi informasi leting 2020, terima kasih selalu memberikan dukungan terbaik sehingga penulis termotivasi dan semangat untuk menyusun tugas akhir ini.
12. Teruntuk seseorang yang akan selalu menjadi salah satu bagian penting dalam hidup penulis. Terima kasih telah menjadi bagian itu dan turut membentuk siapa penulis hari ini.
13. Albiran Nisa, terimakasih atas ketekunan, kesabaran, dan kerja keras yang telah dilakukan selama proses penyusunan tugas akhir ni. Terima kasih telah

tetap berdedikasi meskipun menghadapi berbagai tantangan dan rintangan. Perjalanan ini bukanlah hal yang mudah, tetapi penulis berhasil melewatinya dengan kekuatan doa, usaha dan semangat yang tak pernah padam.

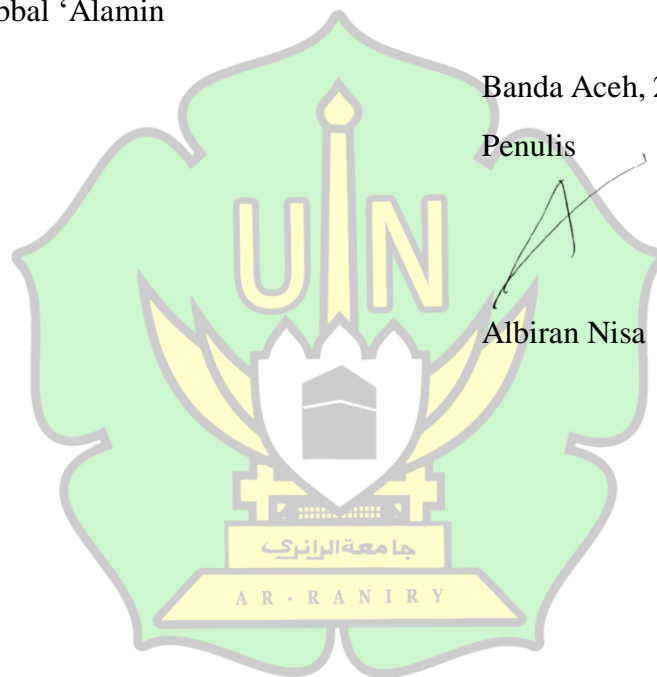
Penulis menyadari sepenuhnya bahwa dalam penyusunan tugas akhir ini tidak cukup dikategorikan sempurna, untuk itu penulis dengan segala kerendahan hati menerima saran dan kritikan guna menyempurnakan penyusunan tugas akhir ini. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi penulis dan pembaca dan semoga dicatat sebagai sebuah amal kebaikan oleh Allah SWT.

Aamiin Ya Rabbal ‘Alamin

Banda Aceh, 20 Agustus 2024

Penulis

Albiran Nisa



DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN	iii
ABSTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Relevan.....	5
2.2 Kerangka Berpikir.....	9
2.3 Landasan Teori.....	10
2.3.1 Implementasi.....	10
2.3.2 Analisis Sentimen.....	10
2.3.3 <i>Artificial Intelligence (AI)</i>	13
2.3.4 <i>Machine Learning</i>	15
2.3.5 <i>Data Mining</i>	15
2.3.6 <i>Text Mining</i>	16
2.3.7 X sebelumnya <i>Twitter</i>	17
2.3.8 Klasifikasi.....	17
2.3.9 <i>Support Vector Machine (SVM)</i>	19
2.3.10 <i>Naive Bayes Classifier</i>	20
2.3.11 <i>Python</i>	21
2.3.12 <i>Google Colaboratory</i>	23
BAB III METODOLOGI PENELITIAN	25
3.1 Tahapan Penelitian.....	25

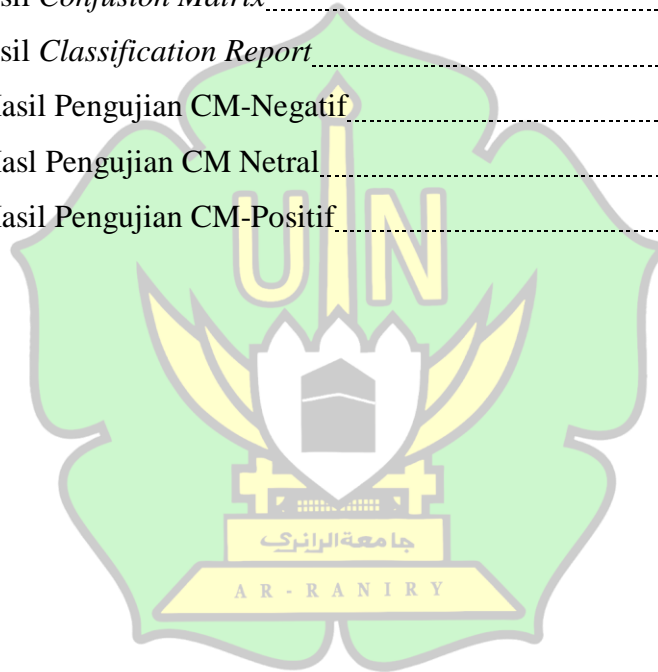
3.1.1 Metode Pengumpulan Data.....	26
3.1.2 <i>Pre-Processing</i> Data.....	28
3.1.3 Pembagian Data.....	30
3.1.4 Implementasi Metode.....	30
3.1.5 Evaluasi Metode.....	32
BAB IV HASIL DAN PEMBAHASAN	34
4.1 Dataset.....	34
4.2 Hasil <i>Pre-Processing</i> Data.....	38
4.3 Pelabelan Data.....	43
4.4 Pembagian Data.....	44
4.5 Pengujian Menggunakan Algoritma SVM.....	46
4.5.1 Pembobotan TF-IDF.....	46
4.5.2 Parameter Model SVM.....	48
4.6 Pengujian Menggunakan Algoritma <i>Naive Bayes</i>	50
4.6.1 Ekstraksi Fitur Menggunakan <i>CountVectorizer</i>	50
4.6.2 Inisialisasi Model <i>Naive Bayes</i>	51
4.7 Evaluasi Performa Model Metode.....	51
4.7.1 Hasil <i>Confusion Matrix</i>	51
4.8 Analisis Hasil Metode.....	56
BAB V KESIMPULAN DAN SARAN	60
5.1 Kesimpulan.....	60
5.2 Saran.....	60
DAFTAR PUSTAKA	62
LAMPIRAN	64
RIWAYAT HIDUP	72

DAFTAR GAMBAR

Gambar II. 1 Kerangka Berpikir.....	9
Gambar III. 1 Tahapan Penelitian.....	25
Gambar III. 2 Tahap <i>Pre-Processing</i>	28
Gambar III. 3 Alur Implementasi Metode.....	31
Gambar IV. 1 <i>Crawling</i> Data pertama.....	35
Gambar IV. 2 Hasil <i>Crawling</i> Data Pertama.....	35
Gambar IV. 3 <i>Crawling</i> Data Kedua.....	35
Gambar IV. 4 Hasil <i>Crawling</i> Data Kedua.....	36
Gambar IV. 5 <i>Crawling</i> Data Ketiga.....	36
Gambar IV. 6 Hasil <i>Crawling</i> Data Ketiga.....	37
Gambar IV. 7 <i>Crawling</i> Data Keempat.....	37
Gambar IV. 8 Hasil <i>Crawling</i> Data Keempat.....	37
Gambar IV. 9 Dataset Awal.....	39
Gambar IV. 10 Hasil Menghapus Data Duplikat.....	39
Gambar IV. 11 Pelabelan Dataset.....	43
Gambar IV. 12 Pembagian Dataset.....	44
Gambar IV. 13 Hasil Data <i>Training</i>	44
Gambar IV. 14 Hasil Data <i>Testing</i>	44
Gambar IV. 15 Hasil <i>Hyperparameter</i>	46
Gambar IV. 16 Hasil Parameter SVM.....	49
Gambar IV. 17 Hasil Ekstraksi Fitur.....	50
Gambar IV. 18 Inisialisasi Model.....	51

DAFTAR TABEL

Tabel IV. 1 Contoh Dataset.....	38
Tabel IV. 2 Proses <i>Cleaning</i>	40
Tabel IV. 3 Proses <i>Case Folding</i>	40
Tabel IV. 4 Proses <i>Tokenization</i>	41
Tabel IV. 5 Proses Normalisasi.....	41
Tabel IV. 6 Proses <i>Filtering</i>	42
Tabel IV. 7 Proses <i>Stemming</i>	42
Tabel IV. 8 Hasil <i>Confusion Matrix</i>	51
Tabel IV. 9 Hasil <i>Classification Report</i>	52
Tabel IV. 10 Hasil Pengujian CM-Negatif.....	56
Tabel IV. 11 Hasil Pengujian CM Netral.....	56
Tabel IV. 12 Hasil Pengujian CM-Positif.....	57



BAB I PENDAHULUAN

1.1 Latar Belakang

Kemajuan perkembangan teknologi saat ini memudahkan pekerjaan serta memenuhi kebutuhan masyarakat termasuk dalam bidang pendidikan khususnya mahasiswa. Salah satu teknologi tersebut adalah *Artificial Intelligence* (kecerdasan buatan) atau disingkat AI merupakan teknologi yang serupa dengan mesin yang dapat meniru perilaku manusia. Ini didasarkan pada pemikiran manusia dan memiliki kemampuan untuk menjalankan proses berpikir manusia. (Goh, 2021). AI atau kecerdasan buatan saat ini, telah mengubah realitas secara mendalam dan memengaruhi cara manusia bekerja, belajar, dan berkomunikasi dalam kehidupan mereka. Kecerdasan buatan mengacu pada kemampuan mesin untuk menunjukkan tingkat kecerdasan dan melakukan berbagai tugas yang memerlukan keterampilan berpikir seperti manusia. Contoh AI *multi-level* yang dapat menangani berbagai tugas dalam kehidupan mahasiswa adalah *ChatGPT*, *Canva*, *Slides Go*, *Grammarly*, *Paraphrase*, dan banyak aplikasi lain yang mendukung penggunaan kecerdasan buatan (Adinda Arly, 2023).

Para mahasiswa saat ini banyak menggunakan *tools* yang telah disediakan untuk membantu mengerjakan berbagai tugas kuliah bahkan digunakan untuk membuat skripsi. Tidak dapat terbantahkan bahwasanya kehadiran AI ini telah mengubah dunia khususnya bagi dunia pendidikan. Namun, seiring meningkatnya penggunaan aplikasi berbasis AI ini menimbulkan ketergantungan pada mahasiswa dengan *tools* yang telah disediakan. Sehingga, mengurangi kemampuan mereka untuk berpikir secara kritis serta mengalami kesulitan dalam memecahkan masalah secara mandiri.

Maka dari itu, penulis memerlukan penelitian untuk menguji sentimen masyarakat mengenai penggunaan AI dalam pembuatan skripsi. Untuk dapat menghitung opini masyarakat digunakan analisis sentimen, yaitu proses menemukan keyakinan seseorang tentang sesuatu yang ditulis di media sosial, baik positif, negatif, atau netral, atau dengan mengekstrak data dari teks atau tulisan

(Herwinsyah, 2022). Banyak metode data *mining* yang dapat digunakan untuk mengukur persepsi masyarakat terhadap tentang penggunaan AI terhadap pembuatan skripsi bagi mahasiswa diantaranya antara lain *Naive Bayes*, *Bayes Network*, *K-Nearest Neighbor*, *Decision Tree*, dan *Support Vector Machine*. Penelitian sebelumnya telah melakukan banyak hal tentang proses pengolahan data yang menggunakan data *mining*, salah satunya adalah studi yang dilakukan oleh Rina Noviana dan Isram Rasal tahun 2023, dengan judul “Penerapan Algoritma *Naive Bayes* dan SVM Untuk Analisis Sentimen *Boy Band* BTS Pada Media Sosial *Twitter*”. Kesimpulan dari penelitian ini menunjukkan bahwa analisis sentimen dengan membandingkan dua algoritma dari *Machine Learning*, yaitu *Naive Bayes* dan SVM memperoleh hasil analisis sentimen positif sebesar 78.8% dan analisis sentimen negatif sebesar 21.2%. Pada analisis hasilnya, algoritma SVM memiliki nilai akurasi 81%, sedangkan algoritma *Naive Bayes* memiliki nilai akurasi 79% (Rina Noviana, 2023).

Data yang digunakan penulis berasal dari pendapat masyarakat media sosial X (yang sebelumnya *twitter*) merupakan *platform* jejaring sosial dan *mikroblogging* yang memungkinkan pengguna mengirim teks informasi hingga 140 karakter pada akun mereka melalui SMS, pesan instan, atau surat elektronik. Banyaknya pengguna media sosial tersebut sering memberikan opini, baik opini positif, negatif, maupun netral. Opini dapat disampaikan secara langsung melalui teks atau melalui kata-kata yang ditafsirkan secara verbal. Sehingga, *text mining* diperlukan untuk menangani data tidak terstruktur dari X . Teknik ini dapat menganalisis data teks semi-terstruktur dan tidak terstruktur, sehingga dapat menyelesaikan masalah seperti analisis sentimen. Pada penelitian ini, penulis menggunakan dua metode untuk analisis sentimen pada penelitian ini, yaitu SVM dan *Naive Bayes*.

SVM merupakan salah satu metode pembelajaran mesin yang paling kuat, memiliki kemampuan untuk menangani dataset dengan dimensi tinggi dan bekerja baik dalam kasus kelas-kelas sulit dipisahkan secara linear. Di sisi lain, *Naive Bayes* menawarkan pendekatan yang lebih sederhana dan cepat, namun tidak kalah efektif dalam situasi tertentu. Dengan membandingkan metode SVM dan *Naive Bayes* pada penelitian ini, penulis berharap dapat memperoleh hasil yang lebih baik dan

dapat menunjukkan bahwa dua metode ini lebih akurat untuk analisis sentimen terhadap penggunaan AI dalam pembuatan skripsi pada media sosial X.

1.2 Rumusan Masalah

Rumusan masalah berdasarkan penjelasan latar belakang di atas, adalah:

1. Bagaimana model implementasi metode SVM dan *Naive Bayes* untuk analisis sentimen penggunaan AI dalam pembuatan skripsi ?
2. Bagaimana hasil perbandingan metode SVM dan *Naive Bayes* untuk penggunaan AI dalam pembuatan skripsi ?

1.3 Tujuan Penelitian

Tujuan dari pengerjaan penelitian ini, adalah:

1. Untuk mengetahui model implementasi metode SVM dan *Naive Bayes* pada analisis sentimen terhadap penggunaan AI dalam pembuatan skripsi.
2. Untuk mengetahui hasil perbandingan metode SVM dan *Naive Bayes* pada analisis sentimen terhadap penggunaan AI dalam pembuatan skripsi.

1.4 Batasan Penelitian

Untuk mencapai tujuan yang diinginkan, maka penulis membuat batasan dalam hal ini, meliputi:

1. Penelitian ini hanya berfokus pada data yang diambil dari media sosial X.
2. Penelitian ini menggunakan dua metode klasifikasi, yaitu SVM dan *Naive Bayes*.
3. Data yang diambil adalah data *tweet* atau komentar para pengguna aplikasi.
4. Analisis sentimen yang digunakan menghasilkan sentimen positif, negatif dan netral.
5. *Python* adalah bahasa pemrograman yang digunakan.

1.5 Manfaat Penelitian

Berdasarkan latar belakang, rumusan masalah, dan tujuan penelitian diatas, maka dapat diperoleh manfaat, yaitu:

1. Dapat mengetahui cara analisis sentimen dengan mengimplementasikan metode SVM dan *Naive Bayes*.

2. Dapat memberikan gambaran hasil analisis sentimen opini masyarakat terhadap penggunaan AI untuk pembuatan skripsi mahasiswa dengan membandingkan metode SVM dan *Naive Bayes*.
3. Memberikan inspirasi untuk penelitian selanjutnya. Penelitian tersebut dapat menggunakan topik yang serupa dengan tetap mencari celah dari penelitian sebelumnya dan dapat ditingkatkan lebih baik lagi.
4. Untuk tugas akhir untuk kelulusan S1 Teknologi Informasi di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.



BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Relevan

Penelitian relevan terdiri dari kumpulan penelitian sebelumnya yang digunakan sebagai referensi utama selama proses pembuatan penelitian baru. Penelitian ini tidak hanya berfungsi sebagai dasar atau bahan acuan, tetapi juga memberikan pemahaman yang lebih mendalam tentang metode penelitian yang digunakan, serta temuan yang telah dicapai sebelumnya. Dengan mempelajari penelitian relevan, peneliti dapat memahami pendekatan-pendekatan penelitian yang digunakan dalam penelitian ini. Penelitian yang relevan juga berfungsi sebagai referensi atau tolak ukur untuk analisis yang akan dilakukan peneliti pada penelitian yang sedang dilakukan. Ini membantu para peneliti menemukan kesalahan atau kelemahan dalam penelitian sebelumnya, sehingga penelitian baru dapat memberikan kontribusi yang lebih besar. Pada penelitian ini, penulis telah memilih beberapa penelitian yang relevan untuk digunakan sebagai referensi. Penelitian ini akan digunakan sebagai landasan untuk mengembangkan metode dan analisis yang lebih baik, serta untuk mendukung argumen yang diajukan.

Pertama, penelitian yang dilakukan oleh Lutfi Budi Ilmawana dan Muhammad Aliyazid Mude pada tahun 2020 berjudul "Perbandingan Metode Klasifikasi *Support Vector Machine* dan *Naive Bayes* untuk Analisis Sentimen pada Ulasan Tekstual di *Google Play Store*" menyajikan analisis mendalam tentang efektivitas dua metode klasifikasi yang umum digunakan dalam proses analisis sentimen. Penelitian ini bertujuan untuk membandingkan akurasi dari kedua metode tersebut, yaitu SVM dan *Naive Bayes*, dalam konteks ulasan tekstual berbahasa Indonesia yang terdapat di *Google Play Store*. Dalam penelitian ini, data ulasan pengguna diambil sebagai sampel untuk dianalisis menggunakan kedua metode klasifikasi tersebut. Hasil yang diperoleh menunjukkan bahwa metode SVM berhasil mencapai tingkat akurasi sebesar 81,46%, sementara metode *Naive Bayes* hanya mencapai akurasi sebesar 75,41%. Perbedaan akurasi ini menunjukkan bahwa SVM lebih unggul dalam mengklasifikasikan sentimen dari ulasan-ulasan

tekstual tersebut. Berdasarkan hasil penelitian ini, peneliti menyimpulkan bahwa metode SVM lebih disarankan untuk digunakan dalam proses analisis sentimen pada ulasan tekstual, terutama untuk ulasan dalam bahasa Indonesia di *platform* seperti *Google Play Store*. Temuan ini penting karena dapat memberikan panduan bagi pengembang aplikasi dan peneliti lain dalam memilih metode yang paling efektif untuk menganalisis sentimen pengguna, yang pada akhirnya dapat membantu dalam pengambilan keputusan bisnis dan peningkatan kualitas layanan (Lutfi Budi Ilmawan, 2020).

Kedua, penelitian yang dilakukan oleh Khoirul Abbi Rokhman dan tim pada tahun 2021, yang berjudul “Perbandingan Metode *Support Vector Machine* dan *Decision Tree* Untuk Analisis Sentimen Review Komentar Pada Aplikasi Transportasi Online”, merupakan sebuah studi yang bertujuan untuk mengevaluasi efektivitas dua metode klasifikasi dalam analisis sentimen. Penelitian ini memanfaatkan data ulasan yang diambil dari *Google Play*, *platform* di mana pengguna aplikasi dapat memberikan ulasan dan penilaian terhadap berbagai aplikasi, termasuk aplikasi transportasi online. Dalam penelitian ini, dua metode klasifikasi yang berbeda, yaitu SVM dan *Decision Tree*, diaplikasikan untuk mengkategorikan sentimen dari ulasan pengguna. Hasil analisis menunjukkan bahwa metode SVM mampu menghasilkan akurasi klasifikasi sebesar 90,20%, sementara metode *Decision Tree* menghasilkan akurasi sebesar 89,80%. Meskipun selisih antara kedua metode ini tidak terlalu signifikan, hasil penelitian ini menunjukkan bahwa metode SVM memiliki kinerja yang lebih unggul dibandingkan dengan metode *Decision Tree* dalam konteks analisis sentimen ulasan pengguna. Hal ini mengindikasikan bahwa SVM dapat lebih efektif dalam menangkap pola-pola kompleks dalam data ulasan, yang sering kali berisi nuansa emosi dan opini yang beragam. Dengan demikian, penelitian ini memberikan kontribusi penting dalam pemilihan metode analisis sentimen yang lebih akurat untuk diterapkan pada data ulasan aplikasi di *platform* digital (Khoirul Abbi Rokhman, 2021).

Ketiga, penelitian yang dilakukan oleh Ulfa Khaira, Reni Aryani, dan Reza Wahyu Hardian pada tahun 2022 dengan judul “Perbandingan Algoritma *Naive*

Bayes Classifier, K-Nearest Neighbor, dan Support Vector Machine Dalam Klasifikasi Karakter Individu Pada Akun *Twitter*” memberikan wawasan yang penting terkait kinerja dari berbagai algoritma klasifikasi dalam konteks analisis data media sosial. Dalam penelitian ini, para peneliti mengevaluasi tiga algoritma populer, yaitu *Naive Bayes Classifier, K-Nearest Neighbor* dan *Support Vector Machine*, untuk menentukan efektivitas masing-masing dalam mengklasifikasikan karakter individu berdasarkan data yang diambil dari akun *Twitter*. Hasil dari penelitian ini menunjukkan bahwa algoritma *Naive Bayes Classifier* mencapai tingkat akurasi sebesar 31,5%, menjadikannya algoritma dengan performa terbaik di antara ketiga metode yang diuji. Di sisi lain, algoritma *K-Nearest Neighbor* hanya mencapai tingkat akurasi sebesar 23,8%, yang menempatkannya sebagai algoritma dengan akurasi terendah dalam penelitian ini. Sementara itu, algoritma SVM berada di tengah-tengah dengan tingkat akurasi sebesar 28,4%. Penelitian ini menggarisbawahi pentingnya memilih algoritma yang tepat dalam analisis data, khususnya dalam konteks klasifikasi karakter individu di *platform* media sosial, di mana hasil yang akurat sangat bergantung pada sifat dan kompleksitas data yang digunakan. Secara keseluruhan, studi ini memberikan kontribusi penting dalam bidang data *science*, khususnya dalam penerapan *machine learning* untuk klasifikasi data berbasis teks di media sosial. Hasil yang diperoleh dapat menjadi acuan bagi peneliti lain dalam memilih algoritma yang sesuai untuk tujuan serupa, serta menyoroti kebutuhan akan pengembangan lebih lanjut dalam peningkatan akurasi algoritma klasifikasi (Aristin Chusnul Khotimah, 2022).

Keempat, penelitian yang dilakukan oleh Ginabila dan Ahmad Fauzi pada tahun 2023 dengan judul "Analisis Sentimen Terhadap Pemutar Musik Online *Spotify* Dengan Algoritma *Naive Bayes* dan *Support Vector Machine*" merupakan sebuah studi yang bertujuan untuk mengevaluasi efektivitas algoritma dalam menganalisis sentimen pengguna terhadap layanan *Spotify*. Dalam penelitian ini, dua algoritma yang umum digunakan dalam klasifikasi teks, yaitu *Naive Bayes* dan SVM, diuji untuk melihat seberapa baik mereka dapat mengkategorikan sentimen positif dan negatif dari ulasan pengguna. Hasil penelitian menunjukkan bahwa algoritma *Naive Bayes* memberikan nilai akurasi sebesar 84,73%, yang berarti

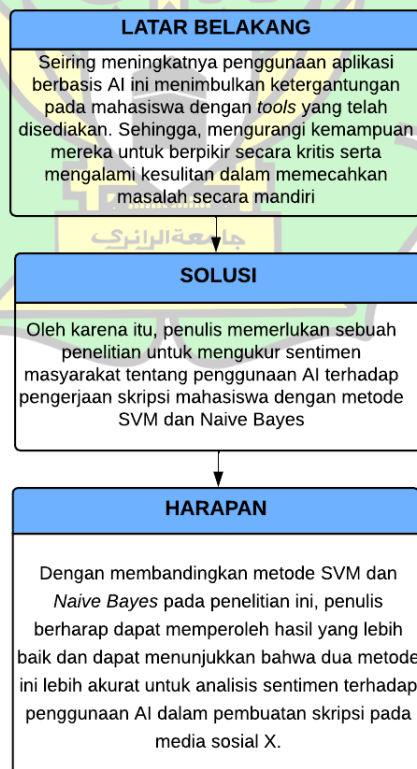
algoritma ini mampu mengklasifikasikan mayoritas ulasan pengguna dengan tepat. Di sisi lain, algoritma SVM meskipun sedikit lebih rendah, juga menunjukkan kinerja yang cukup baik dengan nilai akurasi sebesar 82,42%. Kedua hasil ini mengindikasikan bahwa baik *Naive Bayes* maupun SVM dapat digunakan sebagai alat yang efektif dalam analisis sentimen, namun *Naive Bayes* sedikit lebih unggul dalam konteks penelitian ini. Temuan dari penelitian ini tidak hanya memberikan wawasan mengenai performa algoritma dalam analisis sentimen, tetapi juga menunjukkan potensi untuk diterapkannya metode ini dalam berbagai bidang lain yang membutuhkan pemahaman lebih dalam mengenai opini dan sikap pengguna terhadap suatu produk atau layanan (Ginabila, 2023).

Kelima, penelitian yang dilakukan oleh Aslam Fatkhudin dan timnya pada tahun 2024, dengan judul "*Decision Tree* Berbasis SMOTE dalam Analisis Sentimen Penggunaan *Artificial Intelligence* untuk Skripsi", mengkaji respons publik terhadap penggunaan AI dalam proses penulisan skripsi. Penelitian ini memanfaatkan algoritma *Decision Tree* yang dikombinasikan dengan teknik SMOTE untuk mengatasi ketidakseimbangan data dalam klasifikasi sentimen yang diekstraksi dari media sosial X. Dari hasil analisis, ditemukan bahwa mayoritas pengguna, sebesar 84,4%, memiliki sentimen negatif terhadap penggunaan AI dalam pembuatan skripsi. Hal ini menunjukkan kekhawatiran atau ketidakpuasan yang signifikan dari masyarakat terhadap teknologi ini dalam konteks akademis. Sementara itu, hanya 15,6% dari sentimen yang dianalisis menunjukkan sikap positif, mencerminkan pandangan yang lebih menerima atau mendukung penggunaan AI dalam proses akademik tersebut. Selain itu, penelitian ini juga mengevaluasi performa algoritma yang digunakan, di mana algoritma *Decision Tree* berbasis SMOTE mampu mencapai tingkat akurasi sebesar 66%. Angka ini menunjukkan tingkat keandalan model dalam mengklasifikasikan sentimen, meskipun masih ada ruang untuk peningkatan dalam akurasi dan efektivitas model klasifikasi ini. Penelitian ini memberikan wawasan penting mengenai persepsi masyarakat terhadap integrasi AI dalam dunia akademis, sekaligus menyoroti tantangan yang ada dalam analisis sentimen menggunakan algoritma pembelajaran mesin (Aslam Fatkhudin, 2024).

2.2 Kerangka Berpikir

Kerangka berpikir adalah struktur konseptual yang digunakan untuk mengorganisir dan menghubungkan ide-ide, konsep, teori, serta variabel-variabel yang relevan dalam suatu penelitian atau analisis. Fungsinya adalah sebagai panduan yang membantu peneliti dalam memahami bagaimana berbagai elemen berinteraksi dan saling mempengaruhi, sehingga dapat menjelaskan fenomena atau masalah yang sedang dikaji.

Dalam konteks penelitian, kerangka berpikir merangkum landasan teori dan asumsi-asumsi yang mendasari penelitian, serta menggambarkan hubungan antara variabel independen, variabel dependen, dan variabel-variabel lain yang relevan. Kerangka berpikir ini penting karena memberikan arah yang jelas dalam proses pengumpulan data, analisis, dan interpretasi hasil, sehingga membantu peneliti mencapai kesimpulan yang valid dan reliabel. Kerangka berpikir penulis tuangkan pada gambar II.1



Gambar II. 1 Kerangka Berpikir

2.3 Landasan Teori

2.3.1 Implementasi

Implementasi adalah proses yang melibatkan pelaksanaan atau penerapan rencana, strategi, atau kebijakan yang telah disusun sebelumnya. Proses ini bukan sekadar tindakan sederhana, melainkan mencakup serangkaian langkah yang sistematis dan terorganisir untuk memastikan bahwa apa yang telah direncanakan dapat dijalankan dengan efektif dan efisien.

Dalam konteks organisasi, implementasi biasanya dimulai setelah perencanaan strategis selesai dan persetujuan dari pihak-pihak terkait telah diperoleh. Tahap ini melibatkan alokasi sumber daya, penugasan tanggung jawab, dan pengaturan waktu untuk setiap langkah yang akan diambil. Selain itu, implementasi juga memerlukan pemantauan terus-menerus untuk memastikan bahwa setiap bagian dari rencana berjalan sesuai dengan tujuan yang telah ditetapkan. Lebih jauh lagi, implementasi sering kali membutuhkan adaptasi dan penyesuaian terhadap kondisi yang mungkin berubah selama proses berlangsung. Ini bisa melibatkan penanganan hambatan, penilaian ulang terhadap tujuan, atau modifikasi strategi untuk mencapai hasil yang optimal. Oleh karena itu, implementasi bukan hanya tentang menjalankan rencana, tetapi juga tentang manajemen perubahan, pengambilan keputusan yang cepat, dan koordinasi yang efektif antara berbagai pihak yang terlibat.

Dalam banyak kasus, istilah "implementasi" tidak hanya merujuk pada pelaksanaan tindakan semata, tetapi juga pada seluruh proses yang dilakukan untuk mencapai tujuan tertentu, mulai dari perencanaan awal hingga evaluasi hasil. Kesuksesan implementasi sangat bergantung pada persiapan yang matang, koordinasi yang baik, serta kemampuan untuk menyesuaikan diri dengan situasi yang dinamis.

2.3.2 Analisis Sentimen

Analisis sentimen merupakan suatu aktivitas untuk memeriksa sudut pandang atau pendapat seseorang tentang suatu topik. Tugas utamanya yaitu

mengkategorikan teks atau kalimat dari dokumen, serta kalimat yang memiliki sifat positif, negatif, atau netral.

Analisis sentimen dilakukan melalui penggunaan teknologi *Neuro Linguistic Program* (pemrosesan bahasa alami) dan *Machine Learning* (pembelajaran mesin) untuk melatih perangkat lunak komputer agar dapat menginterpretasikan dan menganalisis teks dengan cara yang sebanding dengan manusia. Perangkat lunak ini menggunakan pendekatan berbasis aturan, pembelajaran mesin, atau kombinasi keduanya, yang disebut hibrida. Setiap metode memiliki kelebihan dan kekurangan. Pendekatan berbasis *machine learning* lebih mudah disesuaikan dan biasanya dapat menangani situasi yang lebih kompleks, tetapi pendekatan berbasis aturan dapat memberikan hasil hampir *real-time*.

Adapun dibawah ini cara kerja analisis sentimen, yaitu :

A. Analisis Sentimen Berbasis Aturan

Perangkat lunak ini dirancang dengan pendekatan berbasis aturan yang cermat untuk mengidentifikasi dan menempatkan kata kunci tertentu dalam blok teks, dengan mempertimbangkan konteks kata atau kelompok kata yang digunakan oleh penulis. Pada intinya, perangkat ini memindai teks menggunakan pengklasifikasi yang dirancang khusus untuk menemukan kata-kata yang terkategori dalam leksikon positif dan negatif. Misalnya, leksikon positif dapat mencakup kata-kata seperti "terjangkau", "cepat", dan "dibuat dengan baik", sementara leksikon negatif mungkin berisi kata-kata seperti "mahal", "lambat", dan "kualitas buruk". Ketika perangkat lunak menemukan kata-kata ini, ia tidak hanya mengidentifikasinya tetapi juga mengevaluasi konteks di sekitarnya untuk menentukan apakah kata tersebut benar-benar menunjukkan sentimen positif atau negatif.

Selain pengklasifikasian kata-kata, perangkat lunak ini juga memiliki kemampuan untuk menghitung skor sentimen secara keseluruhan berdasarkan frekuensi penggunaan kata-kata dalam leksikon tersebut dan dampak emosional yang diasosiasikan dengan setiap kata. Skor ini kemudian memberikan gambaran

umum tentang nada atau sentimen keseluruhan dari teks yang dianalisis, apakah cenderung positif, negatif, atau netral. Dengan kata lain, perangkat lunak ini tidak hanya mengenali kata-kata individual, tetapi juga memprosesnya dalam konteks yang lebih luas untuk memberikan analisis sentimen yang lebih akurat dan bernuansa.

B. Analisis Sentimen *Machine Learning*

Dalam dunia pembelajaran mesin, algoritma memainkan peran penting dalam mengembangkan perangkat lunak yang mampu memahami dan menganalisis sentimen dalam teks. Proses ini melibatkan pelatihan model untuk mengenali emosi yang terkandung dalam blok teks berdasarkan kata-kata yang digunakan dan urutan kemunculannya. Melalui penggunaan algoritma analisis sentimen, pengembang dapat mengajarkan perangkat lunak untuk mendeteksi dan mengklasifikasikan emosi dengan cara yang menyerupai proses pemahaman manusia. Misalnya, perangkat lunak dapat dilatih untuk mengidentifikasi apakah suatu teks memiliki nada positif, negatif, atau netral.

Selama proses pelatihan ini, perangkat lunak secara bertahap "belajar" dari data yang disediakan, sehingga meningkatkan kemampuannya untuk membuat prediksi yang akurat tentang sentimen dalam teks yang baru. Fenomena ini dikenal sebagai "*machine learning*", di mana model terus berkembang dan beradaptasi berdasarkan data baru yang diterimanya.

Dalam konteks analisis sentimen, ada berbagai jenis algoritma yang digunakan untuk mengklasifikasikan teks. Beberapa di antaranya adalah algoritma klasifikasi yang paling umum dan efektif, SVM, *Naive Bayes*, dan *Random Forest*. Algoritma ini dipilih berdasarkan kemampuannya untuk mengolah data teks dengan baik dan menghasilkan prediksi yang handal, membuat mereka menjadi alat yang sangat berharga dalam berbagai aplikasi analisis sentimen. Berikut adalah beberapa algoritma klasifikasi yang paling umum digunakan:

- a) *Regresi Linier* : Algoritma statistik yang menggambarkan nilai (Y) berdasarkan serangkaian fitur (X).

- b) *Naive Bayes* : Algoritma yang menggunakan teorema Bayes untuk mengategorikan kata-kata dalam sebuah blok teks.
- c) *Support Vector Machine* : Algoritma klasifikasi yang cepat dan efisien yang digunakan untuk memecahkan masalah klasifikasi dua kelompok.
- d) *Deep Learning* (pembelajaran mendalam) : Juga dikenal sebagai jaringan neural tiruan. *Deep Learning* adalah teknik *machine learning* tingkat lanjut yang menghubungkan beberapa algoritma untuk meniru fungsi otak manusia.

C. Pendekatan Hibrida

Pendekatan hibrida dalam analisis teks menggabungkan kekuatan *machine learning* dengan teknik berbasis aturan untuk menciptakan solusi yang tidak hanya cepat tetapi juga sangat akurat. Dalam proses ini, *machine learning* digunakan untuk menangani sejumlah besar data secara efisien, sementara aturan-aturan yang ditetapkan secara manual memberikan kerangka kerja yang tepat untuk menafsirkan nuansa dan kompleksitas bahasa yang mungkin tidak dapat dipahami oleh model *machine learning* saja. Hasilnya adalah sistem yang mampu menghasilkan analisis teks yang lebih canggih dan mendalam.

Namun, meskipun pendekatan ini menawarkan keunggulan signifikan dalam hal akurasi dan kecepatan, ada biaya yang harus dibayar. Pendekatan hibrida membutuhkan lebih banyak sumber daya, baik dalam hal waktu maupun keterampilan teknis. Proses pengembangan dan penyempurnaan model *machine learning* memerlukan data latih yang besar dan berkualitas tinggi, serta waktu yang tidak sedikit untuk melatih model tersebut. Selain itu, pengembangan aturan berbasis manual membutuhkan pemahaman mendalam tentang domain tertentu dan kemampuan untuk merancang aturan yang relevan dan efektif. Secara keseluruhan, meskipun pendekatan hibrida dapat memberikan hasil yang superior, ia menuntut investasi yang lebih besar dibandingkan dengan metode lain yang lebih sederhana.

2.3.3 *Artificial Intelligence (AI)*

Sejarah AI dimulai pada tahun 1956 ketika istilah “*Artificial Intelligence*” pertama kali digunakan oleh John McCarthy dan sekelompok peneliti di Konferensi

Darhmouth. AI adalah salah satu cabang ilmu komputer yang mencakup kemampuan mesin untuk melakukan tugas yang mirip dengan seorang manusia disebut kecerdasan buatan.

Di dunia pendidikan AI ini juga sering digunakan oleh para mahasiswa untuk membantu pekerjaannya. Pada saat ini, AI yang paling sering digunakan adalah *chatgpt*, merupakan salah satu bentuk AI yang menjawab pertanyaan yang *user* berikan. Layaknya seperti web berisi informasi, *chatgpt* saat ini dianggap lebih komplit dan relevan oleh seluruh penggunanya. Di samping itu, AI memiliki kelebihan dan kekurangan sebagai berikut :

A. Kelebihan AI

- a) Kemampuan Analitik : AI mampu menganalisis data dengan cepat dan akurat, yang memungkinkan prediksi dan pengambilan keputusan yang lebih baik.
- b) Efisiensi Tinggi : AI dapat menghemat waktu dan biaya dalam banyak industri, seperti manufaktur, perbankan, dan logistik, dengan kecepatan pemrosesan yang tinggi.
- c) Pembelajaran Mandiri : Dengan belajar dari data yang ada dan meningkatkan kinerjanya seiring waktu, sistem AI dapat terus berkembang dan menjadi lebih cerdas.
- d) Pengambilan Keputusan yang Objektif : AI memiliki kemampuan untuk memproses data secara objektif tanpa dipengaruhi oleh emosi atau bias manusia, yang berarti lebih sedikit kesalahan dan lebih akurat dalam pengambilan keputusan.

B. Kekurangan AI

- a) Ketergantungan pada Data : Data yang digunakan untuk pelatihan sangat memengaruhi kinerja AI. Jika data tidak representatif atau terkontaminasi oleh bias, sistem AI dapat menghasilkan hasil yang tidak akurat atau tidak adil.
- b) Keterbatasan Pemahaman Konteks : AI biasanya memiliki keterbatasan dalam memahami konteks yang kompleks. Misalnya,

AI mungkin tidak dapat memahami makna tersembunyi atau nuansa yang terkandung dalam percakapan.

- c) Keamanan dan Privasi : Data yang dikumpulkan dan digunakan oleh sistem AI dapat rentan terhadap serangan atau penyalahgunaan pihak yang tidak berwenang, yang mengancam keamanan dan privasi data Anda.
- d) Penggantian Pekerjaan Manusia : Sistem AI dapat menggantikan pekerjaan yang sederhana dan berulang, mengancam pekerjaan manusia.

2.3.4 Machine Learning

Beberapa ilmuwan matematika seperti Adrien Marie Legendre, Thomas Bayes, dan Andrey Markov pertama kali menggunakan istilah *Machine Learning* pada tahun 1920-an ketika mereka membahas dasar-dasar *machine learning* dan konsepnya.

Machine Learning atau pembelajaran mesin adalah teknik pendekatan dari *Artificial Intelligence* (AI) yang digunakan untuk meniru untuk menggantikan peran manusia dalam melakukan aktivitas untuk memecahkan masalah. Singkatnya, *machine learning* adalah sebuah mesin yang dibuat untuk dapat belajar dan melakukan pekerjaan tanpa arahan dari penggunanya (Agung Wijoyo, 2024).

Machine Learning yang berasal dari disiplin ilmu seperti matematika, statistika, dan data *mining*, memungkinkan mesin untuk menganalisis data dan mempelajari informasi tanpa harus diprogram ulang atau diinstruksikan secara eksplisit. Tergantung pada konteks pembelajaran yang telah dilakukan, *machine learning* juga dapat melakukan berbagai tugas, termasuk memperoleh data dan mempelajari informasi yang ada.

2.3.5 Data Mining

Data *Mining* merupakan proses pengumpulan data berharga dari sejumlah data yang cukup besar. Proses yang digunakan oleh data *mining* adalah teknik yang mencakup statistik, matematika, kecerdasan buatan, dan pembelajaran mesin. Data

mining sering disebut juga *Knowledge Discovery in Database* (KDD) (M. Afdal, 2022).

Secara umum, terdapat beberapa metode yang digunakan untuk melakukan data *mining* diantaranya :

A. *Association*

Metode berbasis aturan yang dikenal sebagai *association* digunakan untuk menemukan hubungan dan korelasi antara variabel dalam set data tertentu. Analisis ini biasanya terdiri dari pernyataan "*if* atau *then*" sederhana. Untuk memahami kebiasaan konsumsi pelanggan, *association* banyak digunakan untuk menemukan hubungan antara produk dalam keranjang belanja. untuk membuat rencana penjualan baru dan sistem rekomendasi yang lebih baik.

B. *Classification*

Klasifikasi adalah metode yang paling umum digunakan dalam data *mining*. Klasifikasi adalah tindakan untuk memprediksi kelas objek.

C. *Regression*

Metode regresi menggunakan analisis variabel independen untuk menjelaskan variabel dependen. Sebagai contoh, anda dapat memprediksi penjualan suatu produk dengan menggunakan korelasi antara tingkat pendapatan rata-rata pelanggan dan harga produk.

D. *Clustering*

Terakhir, pendekatan *clustering* yaitu membagi kumpulan data menjadi beberapa kelompok berdasarkan hal-hal yang serupa. Sebagai contoh, *customer segmentation* membagi pelanggan ke dalam beberapa grup berdasarkan seberapa mirip mereka.

2.3.6 *Text Mining*

Text Mining adalah penemuan baru yang dibuat oleh komputer yang awalnya merupakan informasi yang tidak diketahui, dengan cara mengesktraksi

informasi secara otomatis dari berbagai sumber tertulis. Pola yang ditemukan dalam *text mining* berasal dari teks data yang tidak terstruktur dalam kumpulan dokumen, bukan dari rekaman basis data yang terformat (Retnowati, 2021).

Text mining dapat diterapkan dalam berbagai bidang, seperti bisnis, kesehatan, ilmu sosial, dan keamanan. Di dunia bisnis, *text mining* digunakan untuk memahami umpan balik pelanggan, menganalisis ulasan produk, atau memonitor media sosial untuk menjaga reputasi merek. Di bidang kesehatan, *text mining* membantu dalam mengekstraksi informasi dari catatan medis elektronik untuk penelitian penyakit, pengembangan obat, atau analisis tren kesehatan.

Secara keseluruhan, *text mining* adalah alat yang sangat kuat untuk mengelola dan menganalisis volume besar data teks yang terus bertambah, memberikan wawasan yang dapat digunakan untuk pengambilan keputusan yang lebih baik dan pemahaman yang lebih dalam tentang berbagai fenomena yang diwakili oleh data teks tersebut.

2.3.7 X sebelumnya *Twitter*

X yang sebelumnya dikenal sebagai *Twitter*, adalah *platform* media sosial yang didirikan pada Maret 2006 oleh Jack Dorsey, Noah Glass, Biz Stone, dan Evan Williams. Awalnya, *twitter* berfungsi sebagai layanan *mikroblogging* di mana pengguna dapat mengirim dan membaca pesan terbatas 140 karakter yang dikenal sebagai "*tweet*". Seiring waktu, batas karakter ini diperluas menjadi 280 karakter pada November 2017 untuk memungkinkan pengguna mengekspresikan diri lebih bebas.

Twitter dengan cepat menjadi populer sebagai *platform* untuk berbagi berita, mengikuti selebriti, mengomentari peristiwa terkini, dan berkomunikasi secara *real-time*. Salah satu fitur khas *twitter* adalah kemampuannya untuk memungkinkan percakapan terbuka di mana pengguna dapat berinteraksi dengan siapa pun, baik itu individu, organisasi, atau figur publik, dengan hanya menggunakan "@" diikuti oleh nama pengguna.

2.3.8 Klasifikasi

Satu metode data *mining* yang menggabungkan data ke dalam kelas atau kelompok yang telah ditentukan. yaitu klasifikasi adalah teknik pembelajaran yang diawasi yang membutuhkan data pelatihan berlabel untuk membuat aturan yang mengkategorikan data uji ke dalam kelas atau kelompok yang telah ditentukan. (Aslam Fatkhudin, 2024).

Proses klasifikasi biasanya dimulai dengan pemilihan fitur atau atribut yang relevan dari data. Fitur-fitur ini adalah variabel yang akan digunakan oleh model untuk menentukan kelas atau kelompok dari suatu entitas data. Setelah fitur-fitur ini dipilih, data pelatihan yang sudah berlabel digunakan untuk melatih model klasifikasi. Model ini belajar untuk mengenali pola dan karakteristik dari setiap kelas atau kelompok berdasarkan kombinasi fitur yang ada. Algoritma klasifikasi seperti *Decision Tree*, *Naive Bayes*, *K-Nearest Neighbors*, dan *Support Vector Machine* adalah beberapa contoh dari algoritma yang umum digunakan dalam metode ini.

Setelah model dilatih, langkah berikutnya adalah menguji model tersebut menggunakan data uji, yang juga sering kali berisi label yang benar untuk mengukur akurasi model. Model yang berhasil akan mampu mengkategorikan data uji ke dalam kelas atau kelompok yang benar sesuai dengan yang telah didefinisikan pada data pelatihan. Akurasi model klasifikasi dapat ditingkatkan melalui proses iteratif yang melibatkan pengoptimalan parameter, pemilihan fitur yang lebih baik, untuk mencapai hasil yang lebih baik.

Metode ini sangat berguna karena tidak hanya membantu dalam membuat keputusan berbasis data yang lebih akurat, tetapi juga dalam otomatisasi proses-proses yang sebelumnya memerlukan banyak intervensi manusia. Dengan model klasifikasi yang efektif, organisasi dapat meningkatkan efisiensi operasional, mengurangi biaya, dan meningkatkan kemampuan untuk membuat prediksi yang tepat waktu, yang pada akhirnya dapat meningkatkan daya saing di pasar.

Namun, penting juga untuk dicatat bahwa keberhasilan klasifikasi sangat bergantung pada kualitas data yang digunakan untuk pelatihan. Data yang bias atau tidak representatif dapat menghasilkan model yang tidak akurat, yang pada

gilirannya dapat menyebabkan kesalahan dalam pengambilan keputusan. Oleh karena itu, pengumpulan data yang tepat, pemilihan fitur yang cermat, dan validasi model yang ketat adalah komponen-komponen penting dalam memastikan bahwa metode klasifikasi memberikan hasil yang andal dan bermanfaat.

2.3.9 *Support Vector Machine* (SVM)

Salah satu metode klasifikasi data *mining* adalah *Support Vector Machine* (SVM), yaitu metode klasifikasi yang efektif untuk masalah nonlinier yang pertama kali dibuat oleh Vapnik pada tahun 1992 (Khoirul Abbi Rokhman, 2021). Algoritma SVM berfungsi untuk menemukan *hyperplane* optimal. *Hyperplane* merupakan fungsi yang memiliki kemampuan untuk membedakan dua kelas. Dalam proses ini, SVM akan memaksimalkan jarak antara batas keputusan dan pola pelatihan (Abdusyukur, 2023). Keunggulan algoritma ini termasuk performa data yang sangat baik dengan banyak atribut dan mudah digunakan. Berbeda dengan metode klasifikasi lainnya, SVM memiliki akurasi klasifikasi yang lebih tinggi. (R. Dasmasele, 2022).

Adapun kelebihan dan kekurangan dari metode SVM, sebagai berikut :

A. Kelebihan SVM

- a) Cocok untuk ruang dimensi tinggi.
- b) Efektif untuk kasus dimana jumlah dimensi lebih besar dari jumlah sampel.
- c) Hemat memori, karena menggunakan training point dari fungsi keputusan (*support vector*).
- d) Bekerja relatif baik ketika ada margin pemisahan yang jelas antar kelas.

B. Kekurangan SVM

- a) Algoritma SVM tidak cocok untuk dataset dalam jumlah yang besar karena membutuhkan waktu *training* yang lama.
- b) SVM tidak bekerja dengan baik ketika dataset memiliki lebih banyak *noise* misalnya kelas target terjadi tumpang tindih.
- c) Jika jumlah fitur untuk setiap titik data melebihi jumlah sampe data *training*, SVM akan memiliki performa yang kurang baik.

- d) Karena *Support Vector Classifier* bekerja dengan meletakkan titik data di atas dan di bawah *hyperlane*, tidak ada kejelasan probabilistik yang tinggi.

2.3.10 *Naive Bayes Classifier*

Naive Bayes Classifier merupakan salah satu metode klasifikasi dengan probabilitas sederhana. Metode klasifikasi *Naive Bayes* melibatkan dua tahap yaitu tahap pelatihan dan klasifikasi. Pada tahap penelitian, sampel data diproses sebagai representasi dari data tersebut. Pada tahap klasifikasi, nilai kategori data didasarkan pada term yang ditemukan dalam data yang diklasifikasikan (Rina Noviana, 2023).

Berdasarkan teorema Bayes, metode klasifikasi *Naive Bayes* menggunakan metode probabilitas dan statistik. Metode ini diciptakan oleh ilmuwan Inggris Thomas Bayes dan bertujuan untuk memprediksi peluang di masa depan berdasarkan pengalaman masa lalu. Metode ini dikenal sebagai teorema Bayes. Salah satu ciri utama klasifikator Dr. Naive Bayes ini adalah keyakinan yang sangat kuat (naif) bahwa setiap kondisi atau kejadian independen.

Adapun kelebihan dan kekurangan dari metode *Naive Bayes*, sebagai berikut :

A. Kelebihan *Naive Bayes*

- a) Mudah diimplementasikan, tidak memerlukan banyak waktu komputasi atau pelatihan dan dapat digunakan untuk tugas-tugas terkait klasifikasi biner maupun kelas jamak.
- b) Menangani data yang hilang dengan baik.
- c) Cepat dan *scalable*, karena dapat digunakan untuk pembelajaran cepat dan tugas klasifikasi waktu nyata, dan dapat dengan mudah diparalelkan untuk dijalankan pada beberapa prosesor atau kluster.
- d) Mudah dipahami.
- e) Berkinerja baik dalam klasifikasi teks.
- f) Berfungsi baik dengan kumpulan data kecil.
- g) Fitur yang tangguh terhadap fitur yang tidak relevan. Hal ini karena ia mengasumsikan bahwa semua fitur bersifat independen satu sama

lain, dan ia menghitung probabilitas hasil tertentu berdasarkan ada atau tidaknya setiap fitur secara independen.

- h) Data pelatihan yang dibutuhkan lebih sedikit.
- i) Menangani data kontinu dan diskrit. *Naive Bayes* adalah algoritma serbaguna yang dapat digunakan untuk menganalisis berbagai macam kumpulan data karena dapat menangani data kontinu dan diskrit. Bergantung pada jenis data, ia menggunakan berbagai distribusi probabilitas, termasuk *Gaussian* dan *Multinomial*.

B. Kekurangan *Naive Bayes*

- a) Asumsi independensi.
- b) Kurangnya fleksibilitas.
- c) Kelangkaan data.
- d) Kepekaan terhadap outlier.
- e) Ketidakseimbangan kelas.
- f) Kemampuan terbatas untuk menangkap interaksi antara fitur.
- g) Kemampuan terbatas untuk menangani variabel kontinu.
- h) Bias terhadap fitur dengan frekuensi tinggi.
- i) Kesulitan dalam menangani data hilang.
- j) Kepekaan terhadap pilihan probabilitas sebelumnya.

2.3.11 *Python*

Python adalah salah satu bahasa pemrograman yang sangat populer di kalangan pengembang, terutama dalam bidang ilmu data, pengembangan perangkat lunak, aplikasi web, dan pembelajaran mesin. *Python* telah menjadi pilihan utama bagi banyak profesional karena sifatnya yang serbaguna dan mudah digunakan. Selain itu, *Python* adalah bahasa pemrograman *open-source*, yang berarti gratis untuk digunakan dan didistribusikan, sehingga sangat menarik bagi individu maupun organisasi dengan berbagai kebutuhan pengembangan.

Keunggulan utama *Python* terletak pada sintaksisnya yang bersih dan sederhana, yang memudahkan para pengembang, baik pemula maupun berpengalaman, untuk menulis dan memahami kode dengan lebih cepat dan efisien. Sifat ini tidak hanya mempercepat proses belajar bagi mereka yang baru memulai,

tetapi juga meningkatkan produktivitas pengembang profesional dalam menyelesaikan proyek-proyek yang kompleks.

Python juga memiliki kemampuan integrasi yang sangat baik dengan berbagai jenis sistem operasi dan *platform*, yang membuatnya sangat fleksibel dan dapat digunakan dalam berbagai lingkungan pengembangan. Dengan dukungan komunitas yang luas dan beragam, *Python* terus berkembang dengan cepat, menawarkan berbagai pustaka dan *framework* yang memudahkan pengembangan aplikasi yang canggih dan efisien.

Kesederhanaan dan kekuatan *Python* menjadikannya pilihan utama dalam berbagai industri, termasuk teknologi informasi, ilmu pengetahuan, pendidikan, dan keuangan, menjadikannya salah satu bahasa pemrograman yang paling relevan dan dicari saat ini.

A. Keunggulan *Python*

- a) Mudah dipelajari, *Python* sangat cocok untuk pemula karena sintaksisnya yang mirip dengan bahasa manusia.
- b) Produktivitas tinggi, karena *Python* memiliki sintaksi yang sederhana dan pustaka standar yang besar.
- c) Versalitas. *Python* dapat digunakan untuk berbagai jenis pengembangan, termasuk pengembangan web, data science, kecerdasan buatan, otomatisasi skrip, aplikasi desktop, dan banyak lagi.
- d) Komunitas yang kuat. *Python* memiliki komunitas yang besar dan aktif, yang menyediakan dokumentasi, tutorial, dan bantuan bagi pengembang dari berbagai tingkat keahlian.

B. Keterbatasan *Python*

- a) Kinerja, karena *Python* adalah bahasa interpretative dan dinamis. Kinerjanya bisa lebih lambat dibandingkan bahasa-bahasa yang dikompilasi seperti C atau C++. Namun, ini bisa diatasi dengan menggunakan pustaka yang ditulis dalam C atau pengoptimalan lainnya.
- b) *Mobile Development*. Meskipun *Python* dapat digunakan untuk pengembangan aplikasi *mobile*, itu bukan pilihan utama dalam

industri ini dibandingkan dengan bahasa seperti *Java*, *Swift*, atau *Kotlin*.

- c) *Global Interpreter Lock* (GIL) adalah mekanisme di *Python* yang memungkinkan hanya satu thread yang dapat mengeksekusi *Python bytecode* pada suatu waktu, yang dapat membatasi kinerja pada aplikasi yang sangat bersifat *multi-threading*.

2.3.12 *Google Colaboratory*

Google Colaboratory atau yang lebih dikenal sebagai *Google Colab*, adalah sebuah produk inovatif dari *Google Internal Research* yang dirancang untuk memudahkan pengembangan dan eksekusi kode *Python* secara online. *Google Colab* menawarkan platform berbasis *cloud* yang memungkinkan pengguna untuk menulis, menjalankan, dan berbagi kode *Python* langsung melalui web *browser* tanpa memerlukan instalasi perangkat lunak tambahan. Keunggulan ini menjadikan *Google Colab* sebagai alat yang sangat bermanfaat bagi para pengembang, data *scientist*, dan peneliti yang membutuhkan lingkungan komputasi yang mudah diakses dan fleksibel.

Salah satu fitur utama *Google Colab* yang membuatnya menonjol adalah kemampuannya untuk mendukung kolaborasi secara *real-time*. Fitur ini memungkinkan beberapa pengguna untuk bekerja bersama-sama pada *notebook* yang sama, mirip dengan cara orang-orang dapat bekerja sama di *Google Docs*. Pengguna dapat dengan mudah berbagi *notebook* mereka dengan rekan kerja atau mitra penelitian hanya dengan beberapa klik, membuat proses kolaboratif menjadi lebih efisien dan terstruktur.

Selain itu, karena file *notebook* disimpan langsung di *Google Drive*, pengguna dapat mengaksesnya dari berbagai perangkat kapan saja dan di mana saja. Ini berarti, pekerjaan yang dimulai di satu perangkat dapat dengan mudah dilanjutkan di perangkat lain tanpa kehilangan progres. Fleksibilitas ini tidak hanya mendukung mobilitas, tetapi juga memastikan bahwa pekerjaan akan selalu aman dan terlindungi dari kehilangan data. Kombinasi dari kapabilitas kolaboratif yang kuat, aksesibilitas yang mudah, serta integrasi dengan layanan *Google* lainnya

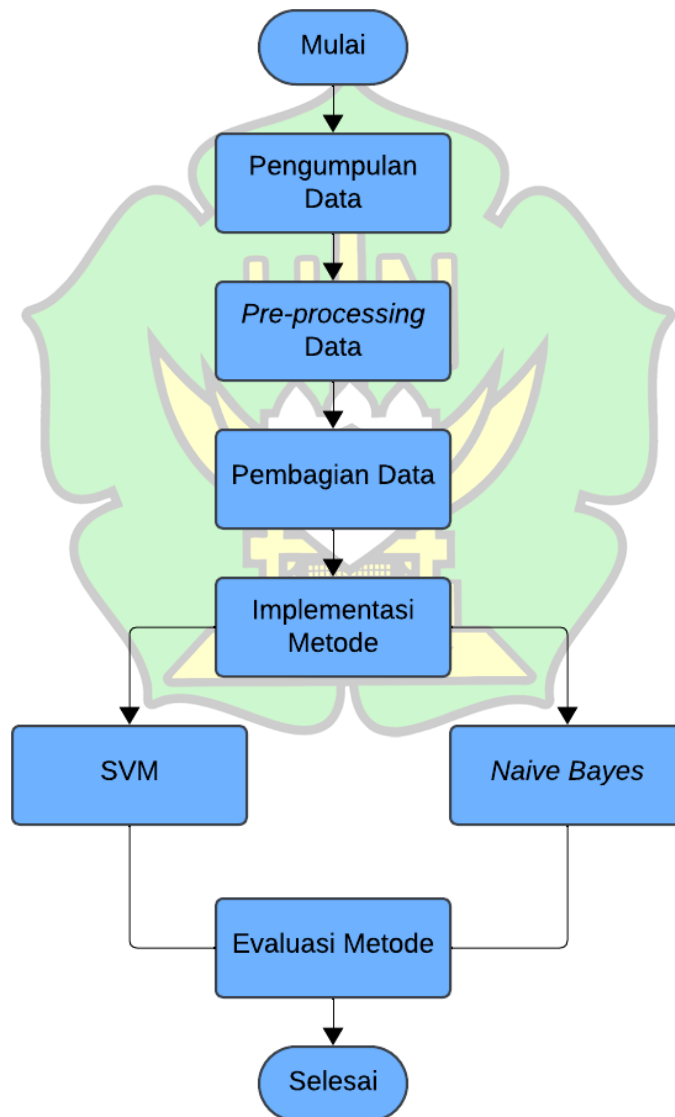
membuat *Google Colab* menjadi alat yang sangat berharga dalam ekosistem pengembangan *Python* berbasis *cloud*.



BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Pada tahapan penelitian ini akan menjelaskan alur penelitian yang digunakan oleh penulis untuk mencapai tujuan penelitian. Tahapan penelitian, seperti terlihat pada gambar III.1 :



Gambar III. 1 Tahapan Penelitian

3.1.1 Metode Pengumpulan Data

Penelitian ini dilakukan dengan menggunakan metode kuantitatif deskriptif yang didukung oleh pendekatan Analisis Data Sekunder (ADS). Metode kuantitatif deskriptif dipilih karena jenis penelitian ini berfokus pada upaya untuk menggambarkan kondisi atau karakteristik variabel penelitian secara objektif, berdasarkan data yang dikumpulkan. Metode ini mengutamakan penggunaan data numerik yang diperoleh dari pengamatan langsung atau pengumpulan data di lapangan, yang kemudian diolah dan dianalisis untuk menghasilkan gambaran yang jelas dan akurat mengenai fenomena yang diteliti.

Penelitian kuantitatif deskriptif menekankan pada deskripsi yang mendalam mengenai variabel-variabel yang ada, tanpa mengarahkan pada hubungan sebab-akibat atau pengujian hipotesis tertentu. Melalui pendekatan ini, peneliti dapat menyajikan data dalam bentuk angka yang mencerminkan situasi yang nyata, memungkinkan analisis statistik yang mendetail untuk memberikan pemahaman yang lebih dalam tentang topik yang diteliti.

Sementara itu, pendekatan Analisis Data Sekunder (ADS) digunakan sebagai strategi untuk melengkapi dan memperkaya data yang dibutuhkan dalam penelitian ini. Data sekunder adalah data yang telah dikumpulkan oleh pihak lain untuk tujuan selain penelitian ini, seperti data yang diperoleh dari laporan penelitian sebelumnya, data dari instansi pemerintah, jurnal, buku, atau sumber-sumber lainnya. Penggunaan data sekunder memungkinkan peneliti untuk memperoleh informasi yang relevan tanpa harus mengumpulkan data baru dari lapangan, yang sering kali memakan waktu dan sumber daya yang lebih besar.

Dengan memanfaatkan Analisis Data Sekunder, penelitian ini tidak hanya mampu menggambarkan variabel-variabel yang ada dengan lebih komprehensif, tetapi juga menghemat waktu dan biaya, karena peneliti dapat mengakses dan memanfaatkan data yang sudah tersedia. Penggunaan data sekunder juga memungkinkan peneliti untuk melakukan analisis perbandingan atau tren dari data historis, yang dapat memberikan wawasan lebih luas terhadap hasil penelitian.

Secara keseluruhan, kombinasi metode kuantitatif deskriptif dengan pendekatan analisis data sekunder dalam penelitian ini diharapkan dapat menghasilkan gambaran yang komprehensif dan mendalam tentang fenomena yang sedang diteliti, dengan dukungan data numerik yang akurat dan relevan dari berbagai sumber yang kredibel.

Pengumpulan data dalam penelitian ini dilakukan dengan memanfaatkan komentar atau *tweet* yang dihasilkan oleh para pengguna aplikasi X. Data tersebut diambil melalui proses *crawling* data yang dilakukan secara otomatis menggunakan bahasa pemrograman *Python*. Proses *crawling* data ini merupakan teknik penting dalam pengumpulan informasi dari situs web, di mana data diekstraksi dari berbagai halaman web secara otomatis, tanpa harus melakukan pengambilan data secara manual yang memakan waktu dan sumber daya.

Crawling data dalam konteks ini bertujuan untuk mengumpulkan data dalam jumlah besar dari komentar atau *tweet* yang dipublikasikan oleh pengguna aplikasi X. Proses ini melibatkan penjelajahan situs web atau *platform* media sosial secara sistematis, mengidentifikasi dan mengunduh konten yang relevan berdasarkan kriteria tertentu. *Python*, dengan berbagai pustaka dan *framework* seperti *BeautifulSoup*, *Scrapy*, dan *Tweepy*, memungkinkan peneliti untuk membangun skrip yang mampu menelusuri dan mengunduh data secara otomatis dari internet. Hal ini sangat berguna dalam penelitian yang membutuhkan data dalam skala besar dan beragam untuk dianalisis.

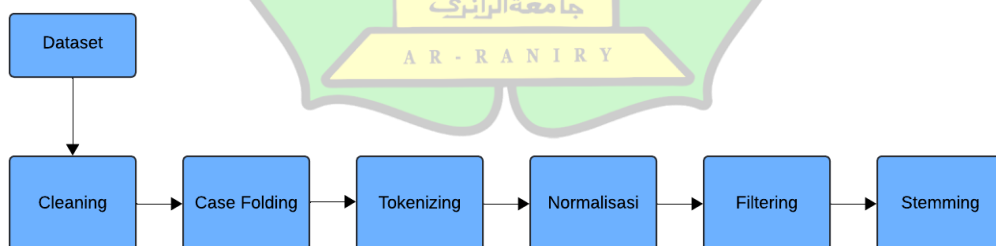
Crawling data tidak hanya sekedar mengumpulkan informasi, tetapi juga memainkan peran penting dalam proses pembuatan indeks data. Indeks ini merupakan kumpulan data terstruktur yang dihasilkan dari proses *crawling*, yang kemudian dapat diolah dan dianalisis lebih lanjut oleh peneliti. Penggunaan *crawling* data dalam pengumpulan data memiliki beberapa keunggulan. Pertama, proses ini memungkinkan pengumpulan data dalam jumlah besar dengan efisiensi tinggi, yang penting dalam penelitian yang membutuhkan dataset yang luas untuk analisis statistik atau pemodelan prediktif. Kedua, *crawling* data dapat dilakukan secara berkala atau *real-time*, sehingga peneliti dapat terus memperbarui data yang

mereka kumpulkan sesuai dengan perkembangan terbaru di *platform* yang diteliti. Ketiga, penggunaan bahasa pemrograman *Python* dalam proses ini menawarkan fleksibilitas dan kemampuan pengolahan data yang kuat, mengingat *Python* memiliki ekosistem pustaka yang luas dan terus berkembang untuk mendukung berbagai kebutuhan pengolahan data.

Secara keseluruhan, pengumpulan data melalui proses *crawling* menggunakan *Python* memberikan peneliti akses yang cepat dan efisien ke data yang dibutuhkan, memungkinkan analisis yang lebih mendalam dan komprehensif terhadap interaksi dan perilaku pengguna di aplikasi X. Proses ini juga berperan penting dalam pembuatan indeks data, yang memfasilitasi pengolahan dan pencarian informasi lebih lanjut, menjadikan *crawling* data sebagai alat yang sangat berharga dalam penelitian berbasis data digital.

3.1.2 *Pre-Processing* Data

Pre-Processing data adalah tahap persiapan yang melakukan pembersihan data. *Pre-Processing* ini mengubah data yang tidak terstruktur menjadi terstruktur, seperti yang diambil dari X. Tahapan *pre-processing* data, seperti terlihat pada gambar III.2 :



.Gambar III. 2 Tahap *Pre-Processing*

- a) *Cleaning* juga dikenal sebagai "pembersihan data", merupakan tahap penting dalam proses *pre-processing* data dalam dunia *data science* dan *machine learning*. Tujuan dari tahap ini adalah untuk membuat data mentah menjadi data yang siap untuk dianalisis lebih lanjut atau dimasukkan ke dalam model *machine learning*. Jika data tidak tersedia, hasil analisis atau model yang dibuat

dapat tidak akurat atau bahkan salah secara total. Jadi, sangat penting untuk memahami dan menerapkan pembersihan data dengan benar. Selain itu, proses *cleaning* juga membantu dalam penghematan waktu dan sumber daya di tahap analisis berikutnya. Data yang bersih lebih mudah dianalisis, menghasilkan interpretasi yang lebih mudah, dan mempermudah pemodelan.

- b) *Case Folding* adalah langkah *pre-processing* yang sederhana namun sangat penting dalam analisis teks. Dengan mengubah semua huruf dalam teks menjadi huruf kecil, *case folding* membantu menyederhanakan proses analisis, menghilangkan potensi duplikasi, dan memastikan bahwa data teks diperlakukan secara konsisten. Meskipun ada beberapa keterbatasan yang perlu diperhatikan, penerapan *case folding* secara umum memberikan manfaat besar dalam banyak aplikasi NLP dan pemrosesan teks. Seperti halnya dengan semua teknik *pre-processing*, penting untuk memahami konteks dan tujuan analisis agar *case folding* diterapkan dengan tepat.
- c) *Tokenizing* adalah proses pemecahan teks menjadi unit-unit yang lebih kecil dan terstruktur, yang dikenal sebagai token. Secara umum, token bisa berupa kata-kata individual, tetapi dalam konteks tertentu, token bisa juga berupa frasa, kalimat, atau bahkan karakter individu.
- d) Normalisasi adalah proses penyeragaman kata yang memiliki makna yang sama. Proses ini bertujuan untuk menyederhanakan teks sehingga lebih mudah untuk di analisis, diproses maupun dibandingkan dengan teks lain.
- e) *Filtering* adalah proses menghilangkan data yang tidak relevan atau yang tidak dibutuhkan dari dataset atau teks. Tujuannya adalah untuk meningkatkan kualitas data dengan memastikan bahwa hanya data yang berguna dan relevan yang dipertahankan. Dalam pemrosesan teks, *filtering* sering kali digunakan untuk menghilangkan kata-kata umum atau *noise*, seperti *stopwords*, tanda baca, angka, atau elemen lain yang dianggap tidak memberikan kontribusi yang signifikan terhadap analisis atau pemodelan.
- f) *Stemming* adalah teknik yang bertujuan untuk mengurangi kata-kata menjadi bentuk dasarnya. Dengan menyederhanakan variasi kata, *stemming* membantu meningkatkan efisiensi, akurasi, dan konsistensi dalam berbagai tugas pemrosesan bahasa alami dan pemrosesan teks. Meskipun ada tantangan dalam

penerapannya, memahami dan menerapkan *stemming* dengan benar adalah kunci untuk menghasilkan analisis teks yang lebih baik dan model yang lebih kuat. Seperti semua teknik *pre-processing*, *stemming* harus digunakan dengan hati-hati dan dengan mempertimbangkan konteks spesifik dari data yang dianalisis.

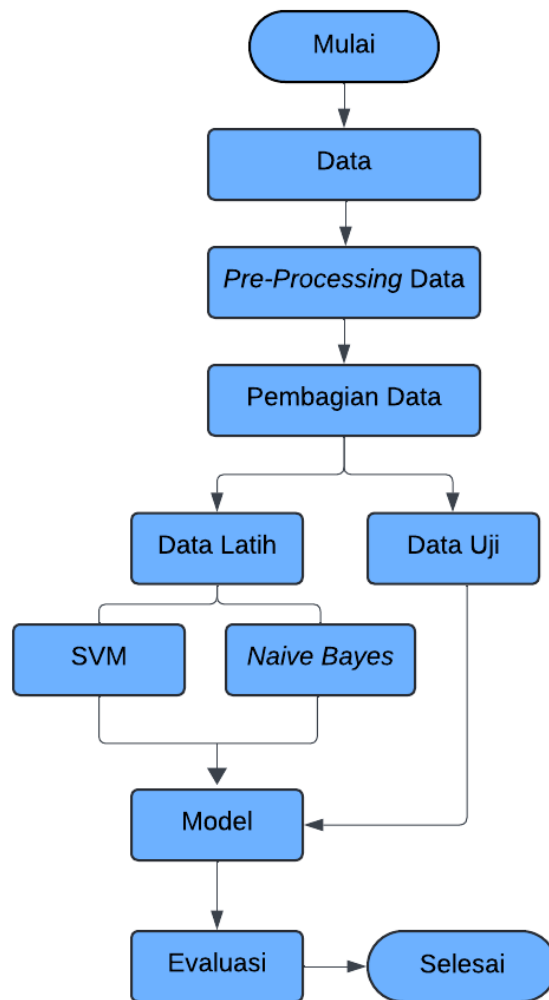
3.1.3 Pembagian Data

Data yang dikumpulkan dalam penelitian ini akan dibagi secara sistematis menjadi dua komponen utama, yang masing-masing akan memainkan peran penting dalam proses analisis dan pengujian:

- a) Data latih, adalah kumpulan data yang akan diproses menggunakan metode atau algoritma yang telah ditentukan untuk penelitian ini. Tujuan pengolahan data latih ini adalah untuk melatih model untuk mempelajari pola-pola dalam data. Hasil dari pengolahan data latih ini akan digunakan sebagai dasar atau referensi untuk membuat prediksi tentang data uji yang belum pernah dilihat oleh model sebelumnya.
- b) Data uji, adalah sekumpulan data yang akan digunakan setelah model dilatih dengan data latih. Data uji ini berfungsi untuk menguji kemampuan model dalam memprediksi hasil dengan tingkat akurasi yang diharapkan. Pada tahap ini, model akan mencoba untuk memprediksi atau mengklasifikasikan data uji berdasarkan pola yang telah dipelajari dari data latih, sehingga dapat dievaluasi seberapa baik model tersebut dalam melakukan prediksi terhadap data baru yang tidak dikenalnya.

3.1.4 Implementasi Metode

Tahap implementasi metode merupakan tahapan untuk memodifikasi dan merencanakan konsep penelitian yang telah dibangun sebelumnya. Pada penelitian ini, penulis menggunakan dua metode, yaitu SVM dan *Naive Bayes*. Alur sistem implementasi metode pada model SVM dan *Naive Bayes* dilihat pada gambar III.3.



Gambar III. 3 Alur Implementasi Metode

Penjelasan dari alur implementasi metode diatas, adalah :

1. Data yang diambil merupakan komentar atau *tweet* para pengguna aplikasi X melalui *tweet-harvest* yaitu *tools* yang digunakan untuk melakukan *crawling* data dengan *Application Programming Interface* (API) pada bahasa pemrograman *python*. Data yang telah dikumpulkan sebanyak **705** data. Data tersebut merupakan *tweet* berbahasa Indonesia dan tersimpan kedalam file dengan format *csv*.
2. *Pre-Processing* adalah tahap pembersihan data yaitu mengubah data yang tidak terstruktur menjadi data terstruktur. Dengan melalui proses *pre-processing* ini

data akan menjadi lebih terstruktur dan menghasilkan nilai yang akurat saat diuji. Beberapa tahapan pada proses *pre-processing* ini adalah *Cleaning*, *Case Folding*, *Tokenizing*, *Normalisasi*, *Filtering*, dan *Stemming*.

3. Data yang digunakan akan dibagi menjadi dua bagian, yaitu data latih dan data uji. Pada umumnya, rasio pembagian data adalah 80:20 (80% merupakan data latih dan 20% merupakan data uji).
4. Model metode *machine learning* pada penelitian ini adalah menggunakan SVM dan *Naive Bayes*. SVM dengan kekuatannya dalam menangani data dengan *margin* yang ketat dan kompleksitas tinggi, sedangkan *Naive Bayes* dengan kemampuannya dalam menangani data secara cepat dan efisien dan dapat menjadi alternatif yang sangat berguna ketika kecepatan dan skalabilitas menjadi prioritas utama.

3.1.5 Evaluasi Metode

Model *machine learning* pada penelitian yang menggunakan metode SVM dan *Naive Bayes* akan di evaluasi menggunakan *confusion matrix*, yaitu alat evaluasi kinerja yang penting dalam klasifikasi data *mining*, serta memberikan gambaran menyeluruh tentang hasil prediksi model. *Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih.

Dalam *confusion matrix*, terdapat 4 istilah nilai hasil klasifikasi yang nantinya akan digunakan untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *f1-score*, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Nilai *True Positive* (TP) dan *True Negative* (TN) adalah hasil klasifikasi yang benar. Nilai *False Positive* (FP) merupakan nilai dimana hasilnya diprediksi positif namun sebenarnya negatif, sedangkan *False Negative* (FN) adalah nilai dimana hasil prediksinya negatif namun sebenarnya positif (Mujaddid Izzul Fikri, 2020).

Berikut adalah rumus dalam perhitungan performa matrik :

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} * 100\% \quad (1)$$

$$Precision = \frac{TP}{TP + FP} * 100\% \quad (2)$$

$$Recall = \frac{TP}{TP + FN} * 100\% \quad (3)$$

$$F1 - Score = 2 * \frac{Precision + Recall}{Precision + Recall} \quad (4)$$

Keterangan :

- *Accuracy* adalah hasil pengujian berdasarkan tingkat pendekatan antara nilai aktual dengan nilai prediksi.
- *Recall* adalah nilai prediksi dalam pernyataan benar antara data yang dimodelkan kedalam metode dengan data asli.
- *Precision* adalah nilai kecocokan data antara data yang dimodelkan kedalam metode dengan data asli (data analisis sentimen) dalam pengklasifikasian.
- *F1-Score* adalah nilai rata-rata antara *precision* dan *recall* yang dibobotkan.



BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini, akan dibahas secara mendetail proses perbandingan antara dua metode yang sering digunakan dalam analisis sentimen, yaitu SVM dan *Naive Bayes*. Perbandingan ini bertujuan untuk mengidentifikasi keakuratan masing-masing metode dalam konteks analisis sentimen. Dalam analisis sentimen, penting untuk memilih metode yang tidak hanya akurat, tetapi juga efisien dalam menangani data yang tersedia. Dengan membandingkan SVM dan *Naive Bayes*, kita dapat mengevaluasi kekuatan dan kelemahan masing-masing pendekatan, serta menentukan metode mana yang paling cocok untuk diterapkan pada data yang telah dikumpulkan. Analisis ini akan melibatkan pengujian terhadap berbagai parameter dan metrik kinerja, sehingga dapat diperoleh pemahaman yang lebih jelas tentang keandalan dan efektivitas kedua metode tersebut dalam menginterpretasikan sentimen dari data. Selain itu, bab ini juga akan mengulas aspek praktis dari penerapan masing-masing metode, termasuk kebutuhan komputasi, kompleksitas model, dan hasil yang diperoleh. Dengan demikian, diharapkan dapat diambil keputusan yang lebih terinformasi mengenai metode analisis sentimen yang optimal untuk kasus yang sedang diteliti.

4.1 Dataset

Pada tahap awal penelitian ini, penulis telah mengumpulkan data melalui proses *crawling* menggunakan *Google Colab* dengan bantuan *tools tweet-harvest*. Proses ini menghasilkan total **712** data, yang diperoleh dengan menerapkan berbagai kata kunci yang relevan. Data yang terkumpul ini merupakan bagian penting dari analisis yang akan dilakukan, karena akan memberikan wawasan awal tentang tren dan pola yang ada dalam media sosial terkait topik yang diteliti. Dalam penjelasan berikut, penulis akan mendetailkan proses pengumpulan data ini serta menjelaskan kata kunci yang digunakan dan bagaimana data tersebut diorganisasikan. Setiap langkah dalam pengumpulan data ini bertujuan untuk memastikan bahwa informasi yang diperoleh adalah representatif dan dapat digunakan untuk analisis lebih lanjut.

1. Data pertama

```
# Crawl Data

filename = 'skripsi.csv'
search_keyword = 'AI skripsi lang:id'
limit = 700

!npx --yes tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" -l {limit} --token {twitter_auth_token}
```

Gambar IV. 1 *Crawling* Data pertama

Gambar IV.1, menampilkan hasil *crawling* data pertama dengan *filename* “skripsi.csv” dan kata kunci “AI skripsi” yang berbahasa Indonesia. Limit yang diminta pada *crawling* data pertama sebanyak “700” data.

```
Your tweets saved to: /content/tweets-data/skripsi.csv
Total tweets saved: 512
[v2.6.1] No more tweets found, please check your search criteria and csv file result

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 1 times, making sure again...

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 2 times, making sure again...

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 3 times, making sure again...

-- Scrolling... (1)[v2.6.1] No more tweets found, please check your search criteria and csv file result

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20)Got 512 tweets, done scrolling...
;
```

Gambar IV. 2 Hasil *Crawling* Data Pertama

Gambar IV.2, menampilkan hasil *crawling* data pertama yang menghasilkan sebanyak 512 data dari limit yang diminta.

2. Data Kedua

```
filename = 'TA 1.csv'
search_keyword = 'ai tugas akhir lang:id'
limit = 100

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "TOP" -l {limit} --token {twitter_auth_token}
```

Gambar IV. 3 *Crawling* Data Kedua

Gambar IV.3, menampilkan hasil *crawling* data kedua dengan *filename* “TA 1.csv” dan kata kunci “ai tugas akhir” yang berbahasa Indonesia. Limit yang diminta pada *crawling* data kedua sebanyak “100” data.

```
Your tweets saved to: /content/tweets-data/TA_1.csv
Total tweets saved: 85

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 1 times, making sure again...

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 2 times, making sure again...

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 3 times, making sure again...

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20)Got 85 tweets, done scrolling...
```

Gambar IV. 4 Hasil *Crawling* Data Kedua

Gambar IV.4, menampilkan hasil *crawling* data pertama yang menghasilkan sebanyak 85 data dari limit yang diminta.

3. Data Ketiga

```
filename = 'TA 2.csv'
search_keyword = 'ai penelitian lang:id'
limit = 100

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "TOP" -l {limit} --token {twitter_auth_token}
```

Gambar IV. 5 *Crawling* Data Ketiga

Gambar IV.5, menampilkan hasil *crawling* data ketiga dengan *filename* “TA 2.csv” dan kata kunci “ai penelitian” yang berbahasa Indonesia. Limit yang diminta pada *crawling* data ketiga sebanyak “100” data.


```
-- Scrolling... (1)

Your tweets saved to: /content/tweets-data/TA_2.csv
Total tweets saved: 100
Got 100 tweets, done scrolling...
```

Gambar IV. 6 Hasil *Crawling* Data Ketiga

Gambar IV.6, menampilkan hasil *crawling* data ketiga yang menghasilkan sebanyak 100 data sesuai dengan limit yang diminta.

4. Data Keempat

```
filename = 'TA 3.csv'
search_keyword = 'mahasiswa tugas akhir ai lang:id'
limit = 100

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "TOP" -l {limit} --token {twitter_auth_token}
```

Gambar IV. 7 *Crawling* Data Keempat

Gambar IV.7, menampilkan hasil *crawling* data keempat dengan *filename* “TA 3.csv” dan kata kunci “mahasiswa tugas akhir ai” yang berbahasa Indonesia. Limit yang diminta pada *crawling* data ketiga sebanyak “100” data.

```
Your tweets saved to: /content/tweets-data/TA_3.csv
Total tweets saved: 22

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 1 times, making sure again...
[v2.6.1] No more tweets found, please check your search criteria and csv file result

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 2 times, making sure again...

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweets found, please check your search criteria and csv file result
Timeout reached 3 times, making sure again...

-- Scrolling... (1)[v2.6.1] No more tweets found, please check your search criteria and csv file result

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20)Got 22 tweets, done scrolling...
```

Gambar IV. 8 Hasil *Crawling* Data Keempat

Gambar IV.8, menampilkan hasil *crawling* data ketiga yang menghasilkan sebanyak 22 data dari limit yang diminta.

- Data yang didapat akan digabung menjadi satu data dalam sebuah *file* csv. Total data yang digabungkan adalah sebanyak **719** data. Pada penelitian ini data yang diambil adalah data yang berasal dari tahun 2010 hingga 2024, sedangkan ada beberapa data yang diperoleh berasal dibawah tahun 2010, maka data tersebut dihapus dan dikurangi sehingga total data nya menjadi **705** data.

4.2 Hasil *Pre-Processing* Data

Tabel IV. 1 Contoh Dataset

<i>Created at</i>	<i>Full text</i>	<i>Location</i>	<i>Username</i>
Friday, Jun 28- 2024	Ai figma emang keren ya gw liat. Gw pernah make AI figjam buat dortir hasil brainstorming gw dan works di skripsi Gokil	Jakarta Selatan, DKI Jakarta	michael_abrhm
Friday, Apr 02- 2010	ai dah nyapeki nian skripsi sikok nih	Hadlramawt Tsani	royhanjml

Dataset pada tabel IV.1 ini mencakup data yang dikumpulkan dari media sosial dalam rentang waktu 2010 hingga 2024. Setiap entri dalam dataset mencatat tanggal dan waktu posting, teks lengkap dari postingan, lokasi geografis asal postingan, dan nama pengguna yang membuat postingan.

Dataset ini memberikan wawasan yang luas tentang bagaimana tren, topik, dan sentimen pengguna media sosial telah berkembang selama lebih dari satu dekade. Dari data ini, dapat di analisis perubahan dalam penggunaan bahasa, topik populer di berbagai lokasi, serta bagaimana teknologi dan aplikasi baru.

Tabel IV.1 juga merupakan contoh dataset yang akan digunakan untuk melakukan tahap *pre-processing*. Sebelum memulai tahap *pre-processing* dalam pengolahan data, sangat penting untuk terlebih dahulu menghapus data duplikat yang mungkin ada dalam dataset. Data duplikat dapat muncul karena berbagai

alasan, seperti penggabungan beberapa sumber data, kesalahan input, atau bahkan akibat proses pengumpulan data yang tidak sempurna. Kehadiran data duplikat ini bisa sangat merugikan, karena dapat mempengaruhi kualitas analisis dan model yang akan kamu buat di tahap-tahap selanjutnya.

Menghapus data duplikat adalah langkah awal yang krusial untuk memastikan bahwa dataset yang kamu gunakan benar-benar representatif dan akurat. Proses menghapus data duplikat juga memberikan keuntungan lain, seperti memperkecil ukuran dataset sehingga memudahkan pengolahan data di tahap-tahap berikutnya.

```
[ ] df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 704 entries, 0 to 703
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   tanggal         704 non-null   object
1   waktu           704 non-null   object
2   username        704 non-null   object
3   quote_count     704 non-null   int64
4   reply_count     704 non-null   int64
5   retweet_count   704 non-null   int64
6   favorite_count  704 non-null   int64
7   full_text       704 non-null   object
dtypes: int64(4), object(4)
memory usage: 44.1+ KB
```

Gambar IV. 9 Dataset Awal

Gambar IV.9, merupakan dataset awal yang berjumlah sebanyak 704 entri.

```
df.drop_duplicates(subset = "full_text", keep = "first", inplace = True)
[ ] df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 694 entries, 0 to 703
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   tanggal         694 non-null   object
1   waktu           694 non-null   object
2   username        694 non-null   object
3   quote_count     694 non-null   int64
4   reply_count     694 non-null   int64
5   retweet_count   694 non-null   int64
6   favorite_count  694 non-null   int64
7   full_text       694 non-null   object
dtypes: int64(4), object(4)
memory usage: 48.8+ KB
```

Gambar IV. 10 Hasil Menghapus Data Duplikat

Gambar IV.10, merupakan hasil menghapus data duplikat menjadi **694** entri.

Setelah proses pembersihan dilakukan, dataset yang awalnya berjumlah **704** entri kini berkurang menjadi **694** entri. Artinya, ada 10 entri yang ternyata

merupakan duplikat dan telah berhasil dihapus. Selain itu, proses penghapusan duplikat ini juga membantu dalam optimalisasi pemrosesan data di tahap-tahap selanjutnya. Dengan dataset yang lebih kecil dan lebih bersih, langkah-langkah seperti *pre-processing* dapat dilakukan lebih cepat dan efisien. Setelah data duplikat dihapus, langkah berikutnya dalam siklus pengolahan data adalah melanjutkan ke tahap *pre-processing*. Tahap ini merupakan salah satu bagian yang paling penting dalam keseluruhan proses analisis data atau pengembangan model *machine learning*. Tahap *pre-processing* meliputi :

a) Proses *Cleaning*

Tabel IV. 2 Proses *Cleaning*

<i>Cleaning</i>	
Teks	Hasil
Ai figma emang keren ya gw liat. Gw pernah make AI figjam buat dortir hasil brainstorming gw dan works di skripsi Gokil	Ai figma emang keren ya gw liat Gw pernah make AI figjam buat dortir hasil brainstorming gw dan works di skripsi Gokil
ai dah nyapeki nian skripsi sikok nih	ai dah nyapeki nian skripsi sikok nih

Tabel IV.2 menunjukkan hasil dari *cleaning*, yaitu untuk menghilangkan semua karakter dari *tweet*, termasuk tanda baca yang ada pada teks sebelumnya.

b) Proses *Case Folding*

Tabel IV. 3 Proses *Case Folding*

<i>Case Folding</i>	
Teks	Hasil
Ai figma emang keren ya gw liat. Gw pernah make AI figjam buat dortir hasil brainstorming gw dan works di skripsi Gokil	ai figma emang keren ya gw liat gw pernah make ai figjam buat dortir hasil brainstorming gw dan works di skripsi gokil
ai dah nyapeki nian skripsi sikok nih	ai dah nyapeki nian skripsi sikok nih

Tabel IV.3 menunjukkan hasil dari *case folding*, yaitu mengubah semua kata menjadi huruf kecil untuk menyeragamkannya. Semua huruf dari a sampai z diterima, kecuali huruf yang dihilangkan.

c) Proses *Tokenization*

Tabel IV. 4 Proses *Tokenization*

Tokenization	
Teks	Hasil
Ai figma emang keren ya gw liat. Gw pernah make AI figjam buat dortir hasil brainstorming gw dan works di skripsi Gokil	ai,figma,emang,keren,ya,gw,liat,gw,pernah,make,ai,figjam,buat,dortir,hasil,brainstorming,gw,dan,works,di,skripsi,gokil
ai dah nyapeki nian skripsi sikok nih	ai,dah,nyapeki,nian,skripsi,sikok,nih

Tabel IV.4 menunjukkan hasil dari *tokenization*, yaitu kata-kata akan dikategorikan dan dibagi menjadi token khusus. Token dapat berupa kata-kata, karakter, atau bagian-bagian kata.

d) Proses Normalisasi

Tabel IV. 5 Proses Normalisasi

Normalisasi	
Teks	Hasil
Ai figma emang keren ya gw liat. Gw pernah make AI figjam buat dortir hasil brainstorming gw dan works di skripsi Gokil	ai figma emang keren ya gw lihat gw pernah make ai figjam buat dortir hasil brainstorming gw dan works di skripsi gokil
ai dah nyapeki nian skripsi sikok nih	ai dah nyapeki nian skripsi sikok nih

Tabel IV.5 menunjukkan hasil dari normalisasi, yaitu kata-kata akan diseragamkan untuk lebih mudah diperiksa, diproses, dan dibandingkan dengan teks lainnya.

e) Proses *Filtering*

Tabel IV. 6. Proses Filtering

<i>Filtering</i>	
Teks	Hasil
Ai figma emang keren ya gw liat. Gw pernah make AI figjam buat dortir hasil brainstorming gw dan works di skripsi Gokil	'figma', 'keren', 'lihat', 'figma', 'dortir', 'hasil', 'brainstorming', 'skripsi'
ai dah nyapeki nian skripsi sikok nih	'skripsi'

Tabel IV.6 menunjukkan hasil dari *filtering*, yaitu pengambilan kata-kata penting dari hasil *tokenizing*. Pada teks yang dilakukan filtering, kata yang dieliminasi adalah “Ai”, ”emang”, ”ya”, ”gw”, ”gokil”, ”dah”, ”nyapeki”, ”nian”, ”nih”.

f) Proses *Stemming*

Tabel IV. 7 Proses Stemming

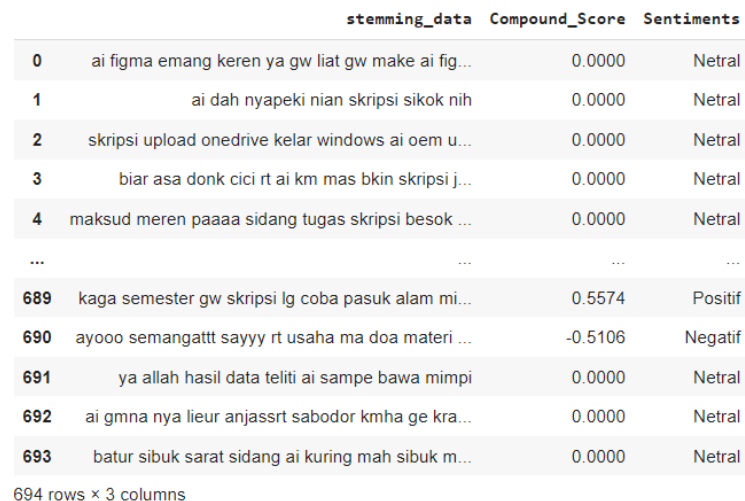
<i>Stemming</i>	
Teks	Hasil
Ai figma emang keren ya gw liat. Gw pernah make AI figjam buat dortir hasil brainstorming gw dan works di skripsi Gokil	figma keren lihat figma dortir hasil brainstorming

	skripsi
ai dah nyapeki nian skripsi sikok nih	skripsi

Tabel IV.7, menunjukkan hasil dari *stemming* yaitu mengembalikan kata berimbuhan ke kata dasar untuk menghindari kesalahan saat menggunakan proses berikutnya.

4.3 Pelabelan Data

Konsep pelabelan data akan dibagi menjadi tiga kelas sentimen yaitu kelas sentimen positif, negatif, dan netral. Jika nilai sentimen dibawah 0 (sentimen < 0) dinyatakan sebagai kelas negatif, jika nilai sentimen di atas 0 (sentimen ≥ 0) dinyatakan sebagai kelas positif, sedangkan nilai sentimen 0 dinyatakan sebagai kelas netral.



	stemming_data	Compound_Score	Sentiments
0	ai figma emang keren ya gw liat gw make ai fig...	0.0000	Netral
1	ai dah nyapeki nian skripsi sikok nih	0.0000	Netral
2	skripsi upload onedrive kelar windows ai oem u...	0.0000	Netral
3	biar asa donk cici rt ai km mas bkin skripsi j...	0.0000	Netral
4	maksud meren paaaa sidang tugas skripsi besok ...	0.0000	Netral
...
689	kaga semester gw skripsi lg coba pasuk alam mi...	0.5574	Positif
690	ayooo semangatt sayyy rt usaha ma doa materi ...	-0.5106	Negatif
691	ya allah hasil data teliti ai sampe bawa mimpi	0.0000	Netral
692	ai gmna nya lieur anjassrt sabodor kmha ge kra...	0.0000	Netral
693	batur sibuk sarat sidang ai kuring mah sibuk m...	0.0000	Netral

694 rows × 3 columns

Gambar IV. 11 Pelabelan Dataset

Berdasarkan gambar IV.11, tabel tersebut menunjukkan hasil analisis sentimen pada data teks yang telah dilakukan *stemming*. Tabel ini memiliki tiga kolom utama:

1. *stemming_data*, berisi teks atau kalimat yang telah diproses menggunakan teknik stemming, yaitu proses mengubah kata-kata menjadi bentuk dasarnya.

2. **Compound_Score**, metrik keseluruhan yang sering digunakan dalam analisis sentimen, terutama ketika menggunakan VADER (*Valence Aware Dictionary and sEntiment Reasoner*), sebuah model berbasis aturan yang dirancang untuk menganalisis sentimen teks, khususnya di media sosial.
3. **Sentiments**, klasifikasi sentimen dari teks yang dihasilkan, yang tampaknya dibagi menjadi tiga kategori: netral, positif, dan negatif.

4.4 Pembagian Data

Dalam konteks pengembangan model *machine learning*, pembagian dataset menjadi data latih dan data uji adalah langkah krusial untuk memastikan bahwa model yang dibangun mampu melakukan generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya. Pada umumnya, rasio pembagian data adalah 80:20 (80% merupakan data latih dan 20% merupakan data uji).

Pembagian data 80:20 antara pelatihan dan pengujian dianggap sebagai praktik umum dalam komunitas pembelajaran *machine learning* dan statistik. Ini dianggap sebagai keseimbangan yang baik antara ukuran data instruksi yang cukup besar untuk memungkinkan model belajar dengan baik dan ukuran data pengujian yang cukup besar untuk menguji kinerja model. Tujuannya adalah untuk memastikan bahwa model tidak *overfitting* (terlalu berfokus pada data pelatihan) dan tetap dapat menggeneralisasi dengan baik ke data baru.

```
[ ] print(f'Jumlah Data Latih: {len(x_train)}')
    print(f'Jumlah Data Uji: {len(x_test)}')
```

⇒ Jumlah Data Latih: 555
Jumlah Data Uji: 139

Gambar IV. 12 Pembagian Dataset

Gambar IV.12 menunjukkan bahwa jumlah data latih yang telah dibagi adalah **555** data, sedangkan jumlah data uji adalah **139** data. Rumusnya dapat hitung seperti ini:

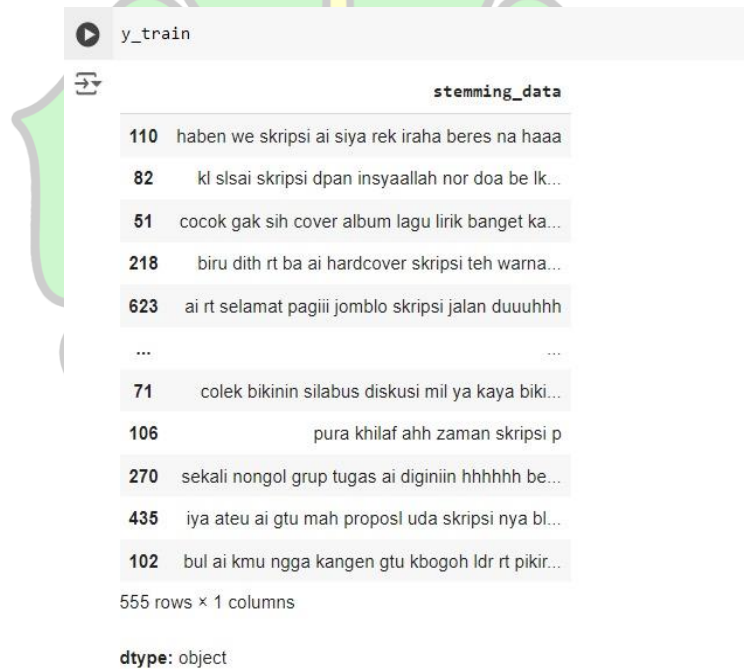
1. Hitung Jumlah Data *Training* (Pelatihan)

$$\begin{aligned} \text{Data Training} &= \text{Total Data} * \text{Persentase Training} \\ &= 694 * 0.8 \\ &= 555.2 = \mathbf{555 \text{ Data}} \end{aligned}$$

2. Hitung Jumlah Data Untuk *Testing* (Pengujian)

$$\begin{aligned} \text{Data Testing} &= \text{Total Data} * \text{Persentase Testing} \\ &= 694 * 0.2 \\ &= 138.8 = \mathbf{139 \text{ Data}} \end{aligned}$$

3. Data *Training* (Pelatihan)



Gambar IV. 13 Hasil Data *Training*

Gambar IV.13, merupakan data *training* yang telah dibagi dengan rasio 80%. Sehingga menghasilkan jumlah sebanyak 555 data sebagai data *training*.

4. Data Testing (Pengujian)

```

y_test
stemming_data
381  mantan mahasiswa tugas libat ai remah peyek ba...
666  samsung buka pusat teliti ai besar inggris htt...
257  sidang skripsi bos ai kd wani umpat race besia...
338  fokus mah skripsi ai maneh hahahha rt geus fok...
319  lg ngerjain skripsi ai selse malm kmu lg rt pe...
...
511  palak kau tenang be selesai gek rt hais paling...
24   jistung nyusun skripsi pake males malesan kaya ...
158  christellasuryo hi tarif implementasi kompetit...
521  ga ai salsa kalo chat gua intro ga bahas skripsi
344  sabar ya sayaanngg rt semangat mengebugebu tug...

139 rows x 1 columns

dtype: object

```

Gambar IV. 14 Hasil Data Testing

Gambar IV.14, merupakan data *testing* yang telah dibagi dengan rasio 20%. Sehingga menghasilkan jumlah sebanyak 139 data sebagai data *testing*.

4.5 Hyperparameter

Hyperparameter adalah parameter yang ditentukan sebelum proses pelatihan model *machine learning* dimulai. Secara keseluruhan, kegunaan *hyperparameter* dalam *machine learning* adalah untuk menjamin bahwa model berfungsi dengan baik dan memenuhi harapan, membantu meningkatkan akurasi model, dan memastikan bahwa *machine learning* dapat digunakan secara efektif untuk memecahkan masalah.

4.5.1 Model Machine Learning

Model *machine learning* yang digunakan untuk tahap *hyperparameter* adalah SVM dan *Naive Bayes*.

Tabel IV. 8 Hasil Hyperparameter

No	Fold	SVM	Naive Bayes
1	1	96.67%	100%
2	2	100%	96.67%
3	3	96.67%	90%
4	4	96.67%	90%

5	5	100%	100%
Rata-Rata		98.00%	95.33%

Tabel IV.8, merupakan hasil dari pengujian *hyperparameter* pada model *machine learning* SVM dan *Naive Bayes* menggunakan *K-Fold Cross Validation* sebanyak 5 *fold*. *K-Fold Cross Validation* adalah teknik evaluasi model yang dapat diimplementasikan menggunakan berbagai *library python*, khususnya *Skicit-Learn* dari *library 'sklearn.model_selection'*. Tujuannya agar melihat performa terbaik yang diperoleh dari masing-masing algoritma.

```
Rata-rata Akurasi SVM (Cross-Validation): 98.00%
Rata-rata Akurasi Naive Bayes (Cross-Validation): 95.33%
[0.96666667 1.          0.96666667 0.96666667 1.          ]
[1.          0.96666667 0.9          0.9          1.          ]
```

Gambar IV. 15 Hasil *Hyperparameter*

a) Proses Detail Untuk Mendapatkan Akurasi Per-Fold Pada SVM

Fold 1: Model SVM dilatih dengan data dari *Fold* 2, 3, 4, dan 5, kemudian diuji pada *Fold* 1. Akurasi yang diperoleh adalah 96.67% atau 0.966666670

Fold 2: Model dilatih dengan data dari *Fold* 1, 3, 4, dan 5, kemudian diuji pada *Fold* 2. Akurasi yang diperoleh adalah 100% atau 1.0

Fold 3: Model dilatih dengan data dari *Fold* 1, 2, 4, dan 5, kemudian diuji pada *Fold* 3. Akurasi yang diperoleh adalah 96.67% atau 0.966666670

Fold 4: Model dilatih dengan data dari *Fold* 1, 2, 3, dan 5, kemudian diuji pada *Fold* 4. Akurasi yang diperoleh adalah 96.67% atau 0.966666670

Fold 5: Model dilatih dengan data dari *Fold* 1, 2, 3, dan 4, kemudian diuji pada *Fold* 5. Akurasi yang diperoleh adalah 100% atau 1.0

b) Proses Detail Untuk Mendapatkan Akurasi Per-Fold Pada Naive Bayes

Fold 1: Model *Naive Bayes* dilatih dengan data dari *Fold* 2, 3, 4, dan 5, kemudian diuji pada *Fold* 1. Akurasi yang diperoleh adalah 100% atau 1.0

Fold 2: Model dilatih dengan data dari *Fold* 1, 3, 4, dan 5, kemudian diuji pada *Fold* 2. Akurasi yang diperoleh adalah 96.67% atau 0.966666670

Fold 3: Model dilatih dengan data dari *Fold* 1, 2, 4, dan 5, kemudian diuji pada *Fold* 3. Akurasi yang diperoleh adalah 90% atau 0.9

Fold 4: Model dilatih dengan data dari *Fold* 1, 2, 3, dan 5, kemudian diuji pada *Fold* 4. Akurasi yang diperoleh adalah 90% atau 0.9

Fold 5: Model dilatih dengan data dari *Fold* 1, 2, 3, dan 4, kemudian diuji pada *Fold* 5. Akurasi yang diperoleh adalah 100% atau 1.0

c) Menghitung Rata-Rata Akurasi

Rata-Rata Akurasi SVM

$$\text{Rata-rata akurasi SVM} = \frac{0.96666667+1.0+0.96666667+0.96666667+1.0}{5} = 0.98 = 98.00\%$$

Rata-Rata Akurasi *Naive Bayes*

$$\text{Rata-rata akurasi Naive Bayes} = \frac{1.0+0.96666667+0.9+0.9+1.0}{5} = 0.9533 = 95.33\%$$

Berdasarkan tabel IV.8, gambar IV.13, dan rumus yang telah digunakan, SVM memiliki *accuracy* tertinggi dengan nilai rata-rata **98.00%**, sedangkan *Naive Bayes* memiliki *accuracy* dengan nilai rata-rata **95.33%**.

4.6 Pengujian Menggunakan Algoritma SVM

4.6.1 Parameter Model SVM

Dalam konteks SVM, terdapat beberapa parameter kunci yang dapat diatur untuk menyesuaikan model agar bekerja optimal pada data yang diberikan. Dengan menggunakan *Support Vector Classifier* (SVC) adalah salah satu implementasi dari algoritma SVM di dalam pustaka '*scikit-learn*', karena SVM adalah algoritma

machine learning yang digunakan untuk tugas klasifikasi yang bekerja dengan menemukan *hyperplane* yang memisahkan kelas-kelas dalam data dengan margin maksimal.

```
Parameter Model SVM:  
Kernel: linear  
C: 1.0  
Intercept : [-0.93022912 -0.6484527  0.81961102]  
Support Vectors:  (0, 5)          0.26058036408017743
```

Gambar IV. 16 Hasil Parameter SVM

Penjelasan dari hasil dari gambar IV.16 :

- a. '**kernel : linear**' parameter ini menentukan jenis kernel yang digunakan oleh SVM. Kernel adalah fungsi yang memetakan data input ke ruang berdimensi lebih tinggi untuk menemukan *hyperplane* yang dapat memisahkan kelas-kelas. Dalam hal ini, *kernel* yang digunakan adalah *linear*, yang berarti SVM akan mencoba menemukan *hyperplane linier* yang terbaik untuk memisahkan data.
- b. '**C : 1.0**' pada parameter SVM menunjukkan pendekatan yang seimbang antara penyesuaian model pada data pelatihan dan pemeliharaan *margin* yang lebih luas. Ini adalah nilai *default* yang umum digunakan dan sering kali berfungsi sebagai titik awal yang baik dalam proses pelatihan model SVM.
- c. '**Intercept**' ini adalah bias atau nilai konstanta yang digunakan dalam fungsi keputusan *hyperplane*. *Intercept* ini adalah titik di mana *hyperplane* memotong sumbu y (dalam kasus sederhana dengan dua dimensi). Ini adalah atribut dari model yang dipelajari selama proses *training*.
- d. '**Support Vectors**' ini adalah vektor-vektor pendukung yang digunakan oleh SVM untuk menentukan *hyperplane* optimal. Vektor-vektor ini adalah titik-titik data yang paling dekat dengan *hyperplane* dan memainkan peran penting dalam menentukan posisi dan orientasi *hyperplane*.

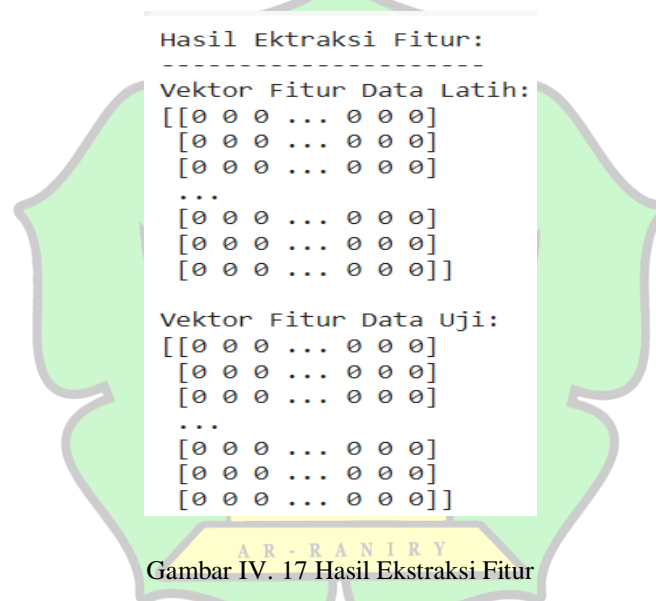
4.7 Pengujian Menggunakan Algoritma *Naive Bayes*

4.7.1 Ekstraksi Fitur Menggunakan *CountVectorizer*

Ekstraksi fitur dalam konteks model *Naive Bayes* biasanya merujuk pada proses mengubah data mentah, seperti teks, menjadi format yang dapat digunakan oleh algoritma untuk pelatihan dan prediksi. Dalam model *Naive Bayes*, sintaks yang diberikan menggunakan *CountVectorizer* dari library '*sklearn.feature_extraction.text*' untuk mengubah teks menjadi matriks hitungan frekuensi kata. Setiap baris dalam matriks mewakili sebuah dokumen, dan setiap kolom mewakili sebuah kata dari kosakata. Nilai di setiap sel adalah frekuensi kemunculan kata dalam dokumen tersebut.

```
Hasil Ekstraksi Fitur:
-----
Vektor Fitur Data Latih:
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

Vektor Fitur Data Uji:
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```



Gambar IV. 17 Hasil Ekstraksi Fitur

Gambar IV.17, menunjukkan hasil ekstraksi fitur menggunakan *CountVectorizer*. Dari gambar tersebut dapat disimpulkan bahwa :

Proses ekstraksi telah berhasil :

- Data latih dan data uji telah berhasil diubah menjadi representasi numerik dalam bentuk vektor fitur.
- Setiap vektor fitur terdiri dari sejumlah elemen (angka 0 dan 1 dalam contoh ini), yang mewakili karakteristik atau fitur yang diekstrak dari data asli.

Struktur data :

- a. Vektor Fitur : Setiap baris dalam matriks mewakili satu sampel data (baik data latih maupun data uji).
- b. Dimensi Fitur : Jumlah kolom dalam matriks menunjukkan jumlah fitur yang diekstrak dari setiap sampel.
- c. Nilai Fitur : Dalam contoh ini, nilai fitur hanya berupa 0 dan 1, yang mungkin mengindikasikan adanya fitur biner (ada atau tidak ada).

4.7.2 Inisialisasi Model *Naive Bayes*

```

MultinomialNB
MultinomialNB()

```

Gambar IV. 18 Inisialisasi Model

Gambar IV.18, menunjukkan model *Multinomial Naive Bayes* (MNB) yang merupakan algoritma klasifikasi probabilistik yang berbasis pada prinsip Bayes dengan asumsi bahwa fitur-fitur independen satu sama lain. Dalam konteks teks, fitur-fitur ini adalah kata-kata atau frasa dalam dokumen. Algoritma ini sangat efisien untuk klasifikasi teks, dan seringkali memiliki banyak kata (fitur) dengan frekuensi yang berbeda-beda.

4.8 Evaluasi Performa Model Metode

Model metode yang digunakan pada penelitian ini adalah algoritma SVM dan *Naive Bayes*, yang telah terbukti efektif dalam berbagai tugas klasifikasi.

4.8.1 Hasil *Confusion Matrix*

Confusion matrix adalah tabel yang menunjukkan distribusi prediksi model terhadap kelas sebenarnya dari data uji.

Tabel IV. 9 Hasil *Confusion Matrix*

<i>Confusion Matrix</i>	
SVM	<i>Naive Bayes</i>
Confusion Matrix untuk SVM: <pre>[[10 0 0] [0 9 0] [0 0 11]]</pre>	Confusion Matrix untuk Naive Bayes: <pre>[[10 0 0] [0 9 0] [0 3 8]]</pre>

Tabel IV.9 merupakan hasil klasifikasi prediksi *confusion matrix* dari metode SVM dan *Naive Bayes*. Sebelum ke tahap klasifikasi, pada tabel SVM, hal pertama yang dilakukan adalah mengimpor fungsi '*confusion_matrix*' dari modul '*metrics*' yang ada didalam *library Scikit-Learn*. Fungsi ini akan digunakan untuk menghasilkan *Confusion Matrix* berdasarkan dua masukan, yaitu label sebenarnya '*y_test*' dan prediksi yang dibuat oleh model '*svm_predictions*'. Setelah menghitung *Confusion Matrix* menggunakan '*confusion_matrix(y_test, svm_predictions)*', hasilnya disimpan dalam variable '*svm_conf_matrix*'.

Pada tabel *Naive Bayes*, sama halnya dengan model SVM, hal pertama yang dilakukan adalah mengimpor fungsi '*confusion_matrix*' dari modul '*metrics*' yang ada didalam *library Scikit-Learn*. Fungsi ini akan digunakan untuk menghasilkan *Confusion Matrix* berdasarkan dua masukan, yaitu label sebenarnya '*y_test*' dan prediksi yang dibuat oleh model *Naive Bayes* '*nb_predictions*'. Setelah menghitung *Confusion Matrix* menggunakan '*confusion_matrix(y_test, nb_predictions)*', hasilnya disimpan dalam variable '*nb_conf_matrix*'.

Tabel IV. 10 Hasil *Classification Report*

Classification Report				
SVM				Naive Bayes
Classification Report untuk SVM:				Classification Report untuk Naive Bayes:
	precision	recall	f1-score	precision recall f1-score su
0	1.00	1.00	1.00	0 1.00 1.00 1.00
1	1.00	1.00	1.00	1 0.75 1.00 0.86
2	1.00	1.00	1.00	2 1.00 0.73 0.84
accuracy			1.00	accuracy 0.90
macro avg	1.00	1.00	1.00	macro avg 0.92 0.91 0.90
weighted avg	1.00	1.00	1.00	weighted avg 0.93 0.90 0.90

Tabel IV.10 menunjukkan hasil *classification report*, yaitu alat evaluasi dalam *machine learning* yang memberikan gambaran lebih rinci tentang performa

model klasifikasi. Berikut penjelasan rinci tentang elemen-elemen yang terdapat pada tabel IV.10, diantaranya :

- a. **'precision'** mengukur seberapa tepat prediksi positif model. Itu dihitung sebagai rasio antara jumlah prediksi positif yang benar dan jumlah total prediksi positif yang dibuat oleh model.
- b. **'recall'** mengukur seberapa baik model dalam menangkap semua contoh positif. Itu dihitung sebagai rasio antara jumlah prediksi positif yang benar dan jumlah total contoh positif yang sebenarnya.
- c. **'f-1 score'** rata-rata harmonis dari *precision* dan *recall*.
- d. **'support'** adalah jumlah *instance* atau contoh aktual dari setiap kelas di dataset uji.
- e. **'accuracy'** adalah nilai akurasi dari keseluruhan model.
- f. **'macro avg'** adalah rata-rata metrik evaluasi (*precision*, *recall*, *f1-score*) untuk semua kelas, dihitung dengan rata-rata sederhana tanpa memperhatikan jumlah *instance* per kelas.
- g. **'weighted average'** adalah rata-rata metrik evaluasi (*precision*, *recall*, *f1-score*) yang memperhitungkan dukungan (jumlah *instance*) untuk setiap kelas, memberikan bobot yang lebih besar pada kelas dengan lebih banyak *instance*.

Berdasarkan *classification report* dari *confusion matrix* yang diberikan, dalam rangka memahami secara lebih mendalam performa masing-masing model, perlu dilakukan perhitungan detail untuk kedua model tersebut, dengan rumus sebagai berikut :

a. *Accuracy*

$$\text{Accuracy} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Semua Prediksi}} * 100\%$$

b. *Precision*

$$\text{Precision} = \frac{TP}{TP + FP} * 100\%$$

c. Recall

$$\text{Recall} = \frac{TP}{TP + FN} * 100\%$$

d. F1-Score

$$F1 - \text{Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

1. Model Metode SVM

a) Akurasi secara keseluruhan untuk model metode SVM, yaitu:

$$\text{Accuracy} = \frac{10 + 9 + 11}{10 + 0 + 0 + 0 + 9 + 0 + 0 + 0 + 11} * 100\%$$

$$\text{Accuracy} = \frac{30}{30} * 100\% = 1.0 * 100\% = 100\%$$

b) Untuk Kelas 0 (Negatif)

TP	FP	FN	TN
10	0	0	9+11 = 20

Perhitungan :

- $\text{Precision} = \frac{10}{10+0} = 1.0 = 100\%$
- $\text{Recall} = \frac{10}{10+0} = 1.0 = 100\%$
- $F1\text{-Score} = 2 * \frac{1.0*1.0}{1.0+1.0} = 1.0 = 100\%$

c) Untuk Kelas 1 (Netral)

TP	FP	FN	TN
9	0	0	10+11 = 21

Perhitungan :

- $\text{Precision} = \frac{9}{9+0} = 1.0 = 100\%$
- $\text{Recall} = \frac{9}{9+0} = 1.0 = 100\%$
- $F1\text{-Score} = 2 * \frac{1.0*1.0}{1.0+1.0} = 1.0 = 100\%$

d) Untuk Kelas 2 (Positif)

TP	FP	FN	TN
11	0	0	10+9 = 19

Perhitungan :

- $Precision = \frac{11}{11+0} = 1.0 = 100\%$
- $Recall = \frac{11}{11+0} = 1.0 = 100\%$
- $F1-Score = 2 * \frac{1.0*1.0}{1.0+1.0} = 1.0 = 100\%$

2. Model Metode Naive Bayes

a) Akurasi secara keseluruhan untuk model metode Naive Bayes, yaitu:

$$Accuracy = \frac{10 + 9 + 8}{10 + 0 + 0 + 0 + 9 + 0 + 0 + 3 + 8} * 100\%$$

$$Accuracy = \frac{27}{30} * 100\% = 0.9 * 100\% = \mathbf{90\%}$$

b) Untuk Kelas 0 (Negatif)

TP	FP	FN	TN
10	0	0	9+3+8 = 20

Perhitungan :

- $Precision = \frac{10}{10+0} = 1.0 = 100\%$
- $Recall = \frac{10}{10+0} = 1.0 = 100\%$
- $F1-Score = 2 * \frac{1.0*1.0}{1.0+1.0} = 1.0 = 100\%$

c) Untuk Kelas 1 (Netral)

TP	FP	FN	TN
9	3	0	10+8 = 18

Perhitungan :

- $Precision = \frac{9}{9+3} = \frac{9}{12} = 0.75 = 75\%$
- $Recall = \frac{9}{9+0} = 1.0 = 100\%$
- $F1-Score = 2 * \frac{0.75*1.0}{0.75+1.0} = \frac{0.75}{1.75} = 0.857 = 85.71\%$

d) Untuk Kelas 2 (Positif)

TP	FP	FN	TN
8	0	3	10+9 = 19

Perhitungan :

- $Precision = \frac{8}{8+0} = 1.0 = 100\%$
- $Recall = \frac{8}{8+3} = \frac{8}{11} = 0.727 = 72.72\%$
- $F1-Score = 2 * \frac{1.0 * 0.727}{1.0 + 0.727} = \frac{0.727}{1.727} = 0.842 = 84.22\%$

4.9 Analisis Hasil Metode

Analisis hasil evaluasi merupakan tahap penting dalam penelitian. Penelitian ini melakukan perbandingan dua metode pada model *Machine Learning* yaitu SVM dan *Naive Bayes* terhadap analisis sentimen penggunaan AI dalam pembuatan skripsi. Hasil dari analisis ini akan memberikan gambaran yang jelas tentang kinerja dari metode-metode. Perbandingan kinerja masing-masing metode diukur dengan *accuracy*, *recall*, *precision* dan *f1-score*.

Tabel IV. 11 Hasil Pengujian CM-Negatif

No	Metode	Jumlah Data Test	Confusion Matrix Kelas 0 (Negatif)			
			Accuracy	Recall	Precision	F1-Score
1	SVM	139	100%	100%	100%	100%
2	Naive Bayes	139	90%	100%	100%	100%

Tabel IV. 12 Hasil Pengujian CM Netral

No	Metode	Jumlah Data Test	Confusion Matrix Kelas 1 (Netral)			
			Accuracy	Recall	Precision	F1-Score
1	SVM	139	100%	100%	100%	100%

2	<i>Naive Bayes</i>	139	90%	100%	75%	85.71%
---	--------------------	-----	-----	------	-----	--------

Tabel IV. 13 Hasil Pengujian CM-Positif

No	Metode	Jumlah Data Test	<i>Confusion Matrix</i> Kelas 2 (Positif)			
			<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
1	SVM	139	100%	100%	100%	100%
2	<i>Naive Bayes</i>	139	90%	72.27%	100%	84.22%

Ketiga tabel diatas memuat hasil pengujian model klasifikasi dua metode *machine learning*, yaitu SVM dan *Naive Bayes*. Pengujian ini dilakukan dengan menggunakan data tes sebanyak 139 data. Ketiga tabel tersebut menguraikan kinerja masing-masing model pada tiap kelas yang terbagi menjadi tiga, yaitu kelas 0 (negatif), kelas 1 (netral), dan kelas 2 (positif). Setiap tabel menunjukkan hasil dari nilai-nilai evaluasi yang meliputi nilai *accuracy*, *recall*, *precision*, dan *f1-score*. Penjelasan untuk ketiga tabel diatas, adalah sebagai berikut :

A. Tabel IV.11 untuk kelas 0 (Negatif)

Pada tabel IV.11, model metode SVM menunjukkan hasil yang sangat sempurna, baik dengan nilai *accuracy*, *recall*, *precision* maupun *f1-score* sebesar 100%., Ini mengindikasikan bahwa model metode SVM mampu mengklasifikasikan semua data dengan akurat dan tepat tanpa ada kesalahan. Sedangkan pada model metode *Naive Bayes*, nilai *accuracy* nya sebesar 90% . Namun, untuk nilai *recall*, *precision*, dan *f1-score*, *Naive Bayes* mencatat nilai yang sempurna sebesar 100%.

B. Tabel IV.12 untuk kelas 1 (Netral)

Pada tabel IV.12, SVM sekali lagi menunjukkan kinerja sempurna dengan semua nilai matrik evaluasi sebesar 100%. Sedangkan, *Naive Bayes* menunjukkan sedikit penurunan performa pada kelas ini dengan *accuracy* yang masih sama sebesar 90%, nilai *recall* sebesar 100%, nilai *precision* sebesar 75%, dan *f1-score* sebesar 85.71%.

C. Tabel IV.13 untuk kelas 2 (Positif)

Pada tabel IV.13, SVM tetap mempertahankan performa sempurna dengan nilai 100% untuk semua metrik evaluasi. Begitupula, *Naive Bayes* dengan performa yang cukup baik meskipun tidak sempurna, dengan *accuracy* yang masih sama sebesar 90%, nilai *recall* sebesar 72.27%, nilai *precision* sebesar 100%, dan nilai *f1-score* sebesar 84.22%..

Dari hasil pengujian yang telah dijelaskan, dapat disimpulkan bahwa metode SVM menunjukkan performa yang sangat sempurna dalam mengklasifikasikan data pada ketiga kelas (negatif, netral, positif). Model metode SVM memiliki nilai *accuracy*, *recall*, *precision*, dan *f1-score* sebesar 100% pada setiap kelas, ini menunjukkan bahwa tidak ada kesalahan klasifikasi yang terjadi pada metode ini. Di sisi lain, pada model metode *Naive Bayes* juga menunjukkan performa yang cukup baik, tetapi tidak sebaik SVM. Pada semua kelas, *Naive Bayes* memiliki nilai *accuracy* yang sama, yaitu 90% dengan sedikit penurunan pada metrik *recall*, *precision*, dan *f1-score* pada kelas netral dan positif. Ini menunjukkan bahwa model metode *Naive Bayes* cenderung kurang presisi dalam mengklasifikasikan beberapa sampel data dibandingkan dengan SVM.

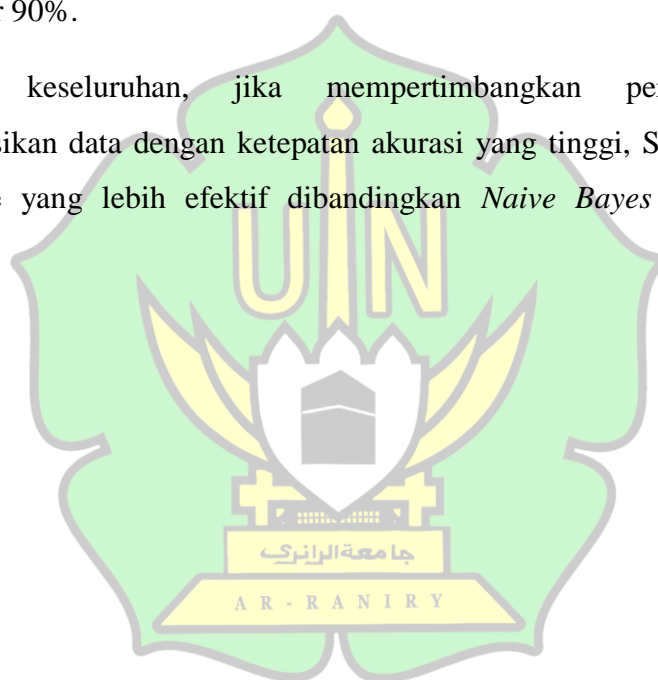
Hasil penelitian ini sejalan dengan penelitian yang dilakukan oleh Rina Noviana dan Isram Rasal tahun 2023, dengan judul “Penerapan Algoritma *Naive Bayes* dan SVM Untuk Analisis Sentimen *Boy Band* BTS Pada Media Sosial *Twitter*”. Perbedaan hasil penelitian yang dilakukan oleh Rina Noviana dan Isram Rasal dengan penelitian ini adalah :

1. Pada penelitian yang dilakukan oleh Rina Noviana dan Isram Rasal tahun 2023, dengan judul “Penerapan Algoritma *Naive Bayes* dan SVM Untuk Analisis Sentimen *Boy Band* BTS Pada Media Sosial *Twitter*”, data yang mereka gunakan

sebanyak **1700** data. Sehingga, algoritma SVM yang digunakan memiliki nilai akurasi 81%, sedangkan algoritma *Naive Bayes* memiliki nilai akurasi 79%. Dan dapat disimpulkan juga, bahwa metode SVM lebih akurat untuk analisis sentimen daripada metode *Naive Bayes* (Rina Noviana, 2023).

2. Pada penelitian yang penulis lakukan dengan judul “Perbandingan Metode *Support Vector Machine* dan *Naive Bayes* Terhadap Penggunaan *Artificial Intelligence* Dalam Pembuatan Skripsi Pada Media Sosial X”, data yang penulis gunakan sebanyak **694** data. Sehingga, menghasilkan nilai akurasi sebesar 100% pada algoritma SVM sedangkan untuk algoritma *Naive Bayes* menghasilkan nilai akurasi sebesar 90%.

Secara keseluruhan, jika mempertimbangkan performa dalam mengklasifikasikan data dengan ketepatan akurasi yang tinggi, SVM merupakan model metode yang lebih efektif dibandingkan *Naive Bayes* dalam konteks pengujian ini.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembahasan pada bab sebelumnya, hasil dari penelitian ini dapat disimpulkan sebagai berikut :

1. Metode pada model penelitian ini adalah SVM dan *Naive Bayes*, yang dapat digunakan dalam proses analisis sentimen melalui rangkaian proses implementasi metode yang terdiri dari pengumpulan data, *pre-processing* data, pembagian data *training* dan data *testing*, implementasi metode *machine learning*, pemodelan, dan evaluasi menggunakan *confusion matrix*.
2. Dari hasil pengujian model klasifikasi menggunakan dua metode *machine learning* pada penelitian ini, dapat disimpulkan bahwa SVM menunjukkan performa yang sangat unggul dengan akurasi 100%. Sedangkan, *Naive Bayes* menunjukkan performa yang cukup baik dengan akurasi 90%. Setelah mengetahui bahwa SVM memberikan akurasi terbaik, lanjutkan fokus pada pengoptimalan parameter SVM lebih dalam untuk mendapatkan hasil yang lebih baik lagi. Meskipun *Naive Bayes* tidak memberikan hasil terbaik, memahami kelemahannya dalam konteks dataset yang digunakan bisa memberikan wawasan penting.

5.2 Saran

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

1. Nilai *accuracy* analisis sentimen dapat dilakukan kembali dengan menggunakan metode –metode klasifikasi lainnya sehingga dapat mengetahui lebih banyak perbedaan tingkat *accuracy*.
2. Penambahan data latih yang lebih beragam, sehingga data yang diperoleh lebih seimbang dan menghasilkan nilai *accuracy* yang lebih akurat.

3. Melakukan analisis mendalam terhadap kesalahan prediksi yang dilakukan oleh model metode klasifikasi sebelumnya, agar dapat mengidentifikasi kelemahan secara spesifik, memahami pola kesalahan yang terjadi, dan mengembangkan strategi perbaikan yang lebih efektif.



DAFTAR PUSTAKA

- Abdusyukur, F. (2023). Penerapan Algoritma Support Vector Machine (SVM) Untuk Klasifikasi Pencemaran Nama Baik di Media Sosial Twitter. *KOMPUTA : jurnal ilmiah komputer dan informatika*, 75.
- Adinda Arly, N. D. (2023). Implementasi Penggunaan Artificial Intelligence Dalam Proses Pembelajaran Mahasiswa Ilmu Komunikasi di Kelas A. *Prosiding Seminar nasional*, 363.
- Agung Wijoyo, A. Y. (2024). Pembelajaran Machine Learning. *OKTAL : Jurnal Ilmu Komputer dan Science*, 376.
- Aristin Chusnul Khotimah, E. U. (2022). Perbandingan Algoritma Naive Bayes Classifier, K-Nearest Neighbor dan Support Vector Machine Dalam Klasifikasi Karakter Individu Pada Akun Twitter. *Jurnal Teknik Informatika (JUTIF)*, 673-680.
- Aslam Fatkhudin, F. A. (2024). Decision Tree Berbasis SMOTE dalam Analisis Sentimen Penggunaan Artificial Intelligence untuk Skripsi. *Remik : Riset dan E-Jurnal Manajemen Informatika Komputer*, 494.
- Dea Oktavia, Y. R. (2023). Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM). *KLIK: Kajian Ilmiah Informatika dan Komputer*, 409.
- Fahlevvi, M. R. (2022). Analisis Sentimen Terhadap Ulasan Aplikasi Pejabat Pengelola Informasi dan Dokumentasi Kementerian Dalam Negeri Republik Indonesia di dan Dokumentasi Kementerian Dalam Negeri Republik Indonesia di Goggle Playstore Menggunakan Support Vector Machine. *Jurnal Teknologi dan Komunikasi Pemerintahan*, 2.
- Ginabila, A. F. (2023). Analisis Sentimen Terhadap Pemutar Musik Online Spotify Dengan Algoritma Naive Bayes dan Support Vector Machine. *Jurnal Ilmiah Ilmu Komputer & Informatika : ILKOMINFO*, 111-122.
- Goh, F. J. (2021). Analisis Yuridids Terhadap Kedudukan Kecerdasan Buatan Atau Artificial Intelligence Sebagai Subjek Hukum Pada Hukum Positif Indonesia. *Supremasi Hukum*, 2.
- Herwinsyah, A. W. (2022). ANALISIS SENTIMEN MASYARAKAT TERHADAP VAKSINASI COVID-19 PADA MEDIA SOSIAL TWITTER MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM). *Jurnal Sistem Informasi dan Informatika (Simika)*, 59.

- Khoirul Abbi Rokhman, B. p. (2021). Perbandingan Metode Support Vector Machine Dan Decision Tree Untuk Analisis Sentimen Review Komentar Pada Aplikasi Transportasi Online. *Jurnal Of Information System Management*, 1-7.
- Luh Putu Ary Sri Tjahyanti, P. S. (2022). Peran Artificial Intelligence (AI) Untuk Mendukung Pembelajaran di Masa Pandemi COVID-19. *Jurnal Komputer dan Teknologi Sains (KOMTEKS)*, 16.
- Lutfi Budi Ilmawan, d. M. (2020). Perbandingan Metode Klasifikasi Support Vector Machine dan Naïve Bayes untuk Analisis Sentimen pada Ulasan Tekstual di Google Play Store. *ILKOM Jurnal Ilmiah*, 154-161.
- M. Afdal, L. R. (2022). Penerapan Text Mining Pada Aplikasi Tokopedia Menggunakan Algoritma K-Nearest Neighbor. *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, 80.
- Mujaddid Izzul Fikri, T. S. (2020). Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter. *SMATIKA JURNAL : STIKI INFORMATIKA JURNAL*, 74.
- Normah, B. R. (2022). Analisa Sentimen Perkembangan Vtuber Dengan Metode Support Vector Machine Berbasis Smote. *Jurnal Teknik Komputer AMIK BSI*, 174-180.
- R. Dasmaselela, B. P. (2022). Penerapan Metode Support Vector Machine (SVM) Untuk Mendeteksi Penyalahan Narkoba. *PARAMETER : Jurnal Informatika, Statistika dan Terapannya*, 112.
- Retnowati, R. S. (2021). Implementasi Pendeteksian Spam Email Menggunakan Metode Text Mining dengan Algoritma Naive Bayes dan Decision Tree J48. *J-ICON : Jurnal Komputer dan Informatika*, 245.
- Rina Noviana, I. R. (2023). Penerapan Algoritma Naive Bayes Dan SVM Untuk Analisis Sentimen Boy Band BTS Pada Media Sosial Twitter. *Jurnal Teknik dan Science*, 51-60.
- Risha Nur Mauliza, Y. R. (2024). Penerapan Text Mining Dalam Menganalisis Pendapat Masyarakat Terhadap Pemilu 2024 Pada Media Sosial X Menggunakan Metode Naive Bayes. *Technomedia Journal (TMJ)*, 1-16.
- Valentino Kevin Sitanayah Que, A. I. (2020). Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 163.

Lampiran

Adapun lampiran lengkap hasil penelitian ini, sebagai berikut :

Lampiran 1. *Source Code* untuk Tahapan *Load Dataset*

Kode dibawah ini digunakan untuk mengimport dataset dan menampilkan dataset.

```
import pandas as pd
data = pd.read_csv('dataset ta albi.csv')
data.info()
```

Lampiran 2. *Source Code* untuk Tahapan *Pre-Processing*

Kode dibawah merupakan proses *Cleaning*

```
import re
import string
import nltk

#Fungsi Untuk Menghapus URL
def remove_URL(tweet):
    if tweet is not None and isinstance(tweet, str):
        # Use re.compile instead of re.compiler
        url = re.compile(r'https?:\/\/\S+www\.\S+')
        return url.sub(r'', tweet)
    else:
        return tweet

#Fungsi Untuk Menghapus HTML
def remove_html(tweet):
    if tweet is not None and isinstance(tweet, str):
        # Use re.compile to compile the regular expression
        html = re.compile(r'<.*?>')
        return html.sub(r'', tweet)
    else:
        return tweet

#Fungsi Untuk Menghapus Emoji
def remove_emoji(tweet):
    if tweet is not None and isinstance(tweet, str):
        # Use Unicode code points to define the emoji ranges
        emoji_pattern = re.compile("[
            u"\U0001F600-\U0001F64F" # emoticons
            u"\U0001F300-\U0001F5FF" # symbols & pictographs
            u"\U0001F680-\U0001F6FF" # transport & map symbols
            u"\U0001F700-\U0001F77F" # alchemical symbols
            u"\U0001F780-\U0001F7FF" # Geometric Shapes Extended
            u"\U0001F800-\U0001F8FF" # Supplemental Arrows-C
            u"\U0001F900-\U0001F9FF" # Supplemental Symbols and
            Pictographs
        ]")
```

```

        u"\U0001FA00-\U0001FA6F" # Chess Symbols
        u"\U0001FA70-\U0001FAFF" # Symbols and Pictographs
Extended-A
        u"\U0001F004-\U0001F0CF" # Additional Emoticons
        u"\U0001F1E0-\U0001F1FF" # Flags
    "]" +", flags=re.UNICODE)
    return emoji_pattern.sub(r'', tweet) # Remove matched
emojis
else:
    return tweet
#Fungsi Untuk Menghapus Simbol
def remove_symbols(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'^a-zA-Z0-9\s]', '', tweet) # Menghapus
semua simbol
    return tweet
#Fungsi Untuk Menghapus Angka
def remove_numbers(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'\d', '', tweet) # Menghapus semua angka
    return tweet
def remove_username(text):
    import re
    return re.sub(r'@[^\s]+', '', text)
df['cleaning'] = df['full_text'].apply(lambda x:
remove_URL(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_username(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_html(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_emoji(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_symbols(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_numbers(x))
df.head(704)

```

Kode dibawah merupakan proses *Case Folding*

```

def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower() # Indent this line
        return lowercase_text
    else:
        return text
df['case_folding'] = df['cleaning'].apply(case_folding)

```

```
df.head(704)
```

Kode dibawah merupakan proses *Tokenization*

```
def tokenize(text):  
    tokens = text.split()  
    return tokens  
df['tokenize'] = df['case_folding'].apply(tokenize)  
df.head(704)
```

Kode dibawah merupakan proses Normalisasi

```
!pip install nltk  
import nltk  
import pandas as pd  
import re  
import string  
from nltk.corpus import stopwords  
from nltk.tokenize import word_tokenize  
  
nltk.download('stopwords') # Download the stopwords data  
  
# Define a new function to handle tokenized data if necessary  
def further_process_tokens(tokens):  
    # Do something with the list of tokens, for example:  
    return [token for token in tokens if token not in  
stopwords.words('indonesian')]  
  
# Apply the new function to the 'tokenize' column  
df['normalisasi'] =  
df['tokenize'].apply(further_process_tokens)  
df.head(704)
```

Kode dibawah merupakan proses *Filtering*

```
!pip install nltk  
from nltk.corpus import stopwords  
nltk.download('stopwords')  
stop_word = stopwords.words('indonesian')  
def remove_stopwords(text):  
    return [word for word in text if word not in stop_word] #  
Use 'stop_word' instead of 'stop_words'  
  
df['stopword removal'] = df['normalisasi'].apply(lambda x:  
remove_stopwords(x))  
df.head(704)
```

Kode dibawah merupakan proses *Stemming*

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

```

from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(text):
    return [stemmer.stem(word) for word in text]
df['stemming_data'] = df['stopword removal'].apply(lambda x: '
'.join(stem_text(x)))
df.head(704)

```

Lampiran 3. Source Code untuk Pelabelan Data

```

!pip install VaderSentiment
from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()
df['stemming_data'] = df['stemming_data'].fillna('')
analyser = SentimentIntensityAnalyzer()
scores = [analyser.polarity_scores(x) for x in
df['stemming_data']]
df['Compound_Score'] = [x['compound'] for x in scores]

df.loc[df['Compound_Score'] > 0, 'Sentiments'] = 'Positif'
df.loc[df['Compound_Score'] < 0, 'Sentiments'] = 'Negatif'
df.loc[df['Compound_Score'] == 0, 'Sentiments'] = 'Netral'
data.head(694)

```

Lampiran 4. Source Code untuk Tahapan Split Data

```

import pandas as pd
from sklearn.model_selection import train_test_split

# Contoh data menggunakan pandas DataFrame
data = pd.read_csv('Hasil-Label-Data-2.csv')

# Misalkan kolom terakhir adalah label
X = data.iloc[:, :-1] # Semua kolom kecuali yang terakhir
sebagai fitur
y = data.iloc[:, -1] # Kolom terakhir sebagai label

# Membagi data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

# Menampilkan ukuran masing-masing subset
print("Jumlah data training:", X_train.shape[0])
print("Jumlah data testing:", X_test.shape[0])

```

Lampiran 5. *Source Code* untuk Tahapan Klasifikasi Data dan Menampilkan Nilai Akurasi pada Metode SVM dan *Naive Bayes*

```
!pip install scikit-learn
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split,
cross_val_score
from sklearn.datasets import load_iris
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

# Memuat dataset Iris
data = load_iris()
X = data.data
y = data.target

# Membagi data menjadi training dan testing set
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Membuat dan melatih model SVM
svm_model = SVC(kernel='linear', random_state=42)
svm_model.fit(X_train, y_train)

# Membuat dan melatih model Naive Bayes
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)

#Memberi label
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(['negatif', 'netral',
'positif'])

# Cek hasil encoding
print(label_encoder.classes_) # Output: ['negatif' 'netral'
'positif']
print(y) # Output mungkin [0, 1, 2] atau dalam urutan lain

# Cross-validation untuk SVM
```



```

from sklearn.model_selection import KFold, cross_val_score
svm_cv_scores = cross_val_score(svm_model, X, y, cv=5)
#prepare cv
cv = KFold(n_splits=5, shuffle=True, random_state=42)
print(f'Rata-rata Akurasi SVM (Cross-Validation):
{np.mean(svm_cv_scores) * 100:.2f}%')

# Cross-validation untuk Naive Bayes
nb_cv_scores = cross_val_score(nb_model, X, y, cv=5)
#prepare cv
cv = KFold(n_splits=5, shuffle=True, random_state=42)
print(f'Rata-rata Akurasi Naive Bayes (Cross-Validation):
{np.mean(nb_cv_scores) * 100:.2f}%')

print(svm_cv_scores)
print(nb_cv_scores)

# Prediksi dengan model SVM
svm_predictions = svm_model.predict(X_test)

# Prediksi dengan model Naive Bayes
nb_predictions = nb_model.predict(X_test)

# Hitung akurasi
svm_accuracy = accuracy_score(y_test, svm_predictions)
nb_accuracy = accuracy_score(y_test, nb_predictions)

print(f'Akurasi SVM: {svm_accuracy * 100:.2f}%')
print(f'Akurasi Naive Bayes: {nb_accuracy * 100:.2f}%')

# Evaluasi tambahan
print('\nConfusion Matrix untuk SVM:')
print(confusion_matrix(y_test, svm_predictions))

print('\nClassification Report untuk SVM:')
print(classification_report(y_test, svm_predictions))

print('\nConfusion Matrix untuk Naive Bayes:')
print(confusion_matrix(y_test, nb_predictions))

print('\nClassification Report untuk Naive Bayes:')
print(classification_report(y_test, nb_predictions))

import numpy as np

# Menghitung confusion matrix

```

```

svm_conf_matrix = confusion_matrix(y_test, svm_predictions) #
Use y_test from the train/test split
print('Confusion Matrix untuk SVM:')
print(svm_conf_matrix)

# Confusion matrix
cm = np.array([[19, 0, 0],
               [ 0, 13, 0],
               [ 0, 0, 13]])

# Inisialisasi dictionary untuk menyimpan TP, TN, FP, FN per
kelas
TP = {}
TN = {}
FP = {}
FN = {}

# Menghitung TP, TN, FP, FN untuk setiap kelas
for i in range(len(cm)):
    TP[i] = cm[i, i] # True Positives untuk kelas i
    FP[i] = sum(cm[:, i]) - cm[i, i] # False Positives untuk
kelas i
    FN[i] = sum(cm[i, :]) - cm[i, i] # False Negatives untuk
kelas i
    TN[i] = sum(sum(cm)) - (TP[i] + FP[i] + FN[i]) # True
Negatives untuk kelas i

# Menampilkan hasil untuk setiap kelas
for i in range(len(cm)):
    print(f"\nKelas {i}:")
    print(f"True Positives (TP): {TP[i]}")
    print(f"True Negatives (TN): {TN[i]}")
    print(f"False Positives (FP): {FP[i]}")
    print(f"False Negatives (FN): {FN[i]}")

import numpy as np

# Confusion Matrix untuk Naive Bayes
nb_conf_matrix = confusion_matrix(y_test, nb_predictions)
print('\nConfusion Matrix untuk Naive Bayes:')
print(nb_conf_matrix)

# Confusion matrix
cm = np.array([[19, 0, 0],
               [ 0, 12, 1],
               [ 0, 1, 12]])

```

```

# Inisialisasi dictionary untuk menyimpan TP, TN, FP, FN per
kelas
TP = {}
TN = {}
FP = {}
FN = {}

# Menghitung TP, TN, FP, FN untuk setiap kelas
for i in range(len(cm)):
    TP[i] = cm[i, i] # True Positives untuk kelas i
    FP[i] = sum(cm[:, i]) - cm[i, i] # False Positives untuk
kelas i
    FN[i] = sum(cm[i, :]) - cm[i, i] # False Negatives untuk
kelas i
    TN[i] = sum(sum(cm)) - (TP[i] + FP[i] + FN[i]) # True
Negatives untuk kelas i

# Menampilkan hasil untuk setiap kelas
for i in range(len(cm)):
    print(f"\nKelas {i}:")
    print(f"True Positives (TP): {TP[i]}")
    print(f"True Negatives (TN): {TN[i]}")
    print(f"False Positives (FP): {FP[i]}")
    print(f"False Negatives (FN): {FN[i]}")

```



RIWAYAT HIDUP



Albiran Nisa, lahir di Tanjung Pura, Sumatera Utara. Pada tanggal 04 Desember 2002. Anak pertama dari tiga bersaudara dari pasangan Ibu Nurmala Sari dan Bapak Syaiful Azmi. Penulis menyelesaikan pendidikan Sekolah Dasar di SDN 050727 Tanjung Pura. Kemudian melanjutkan pendidikan di jenjang Sekolah Menengah Pertama di MTsN Tanjung Pura. Penulis menempuh pendidikan di jenjang Sekolah Menengah Atas di MAN 2 Langkat dan lulus pada tahun 2020. Pada tahun yang sama penulis terdaftar

Sebagai mahasiswi pada program studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh melalui jalur SNMPTN

