

**ANALISIS SENTIMEN KEBIJAKAN
PROGRAM PEMAHADAN TAHUN 2023
UIN WALISONGO SEMARANG DENGAN METODE
*NAIVE BAYES CLASSIFIER***

SKRIPSI

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi
Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1)
dalam Teknologi Infomasi



Diajukan Oleh:
VIORELLA AMANDA PUTRI
NIM: 2008096025

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG
2024**

PERNYATAAN KEASLIAN

Yang bertandatangan di bawah ini :

Nama : Viorella Amanda Putri
NIM : 2008096025
Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul :

**Analisis Sentimen Kebijakan Program Pema'hadan
Tahun 2023 Uin Walisongo Semarang Dengan Metode
*Naïve Bayes Classifier***

Secara keseluruhan adalah hasil penelitian/karya saya kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 26 Maret 2024



ernyataan,

Viorella Amanda Putri

NIM: 2008096025



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI

Alamat: Jl. Prof. Dr. Hamka Km 1, Ngaliyan Semarang Telp.024-76433366 Semarang 50185
Email: fst@walisongo.ac.id, Web: <http://fst.walisongo.c.id>

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Analisis Sentimen Kebijakan Program Pemahadan
Tahun 2023 UIN Walisongo Semarang dengan
Metode *Naive Bayes Classifier*

Nama : **Viorella Amanda Putri**

NIM : 2008096025

Jurusan : Teknologi Informasi

Telah diujikan dalam sidang tugas akhir oleh Dewan
Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang
dan dapat diterima sebagai salah satu syarat memperoleh gelar
sarjana dalam bidang ilmu teknologi informasi.

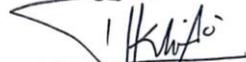
Semarang, Mei 2024

DEWAN PENGUJI

Penguji I


Nur Cahyo Hendro Wibowo, S.T., M.Kom
NIP. 197312222006041001

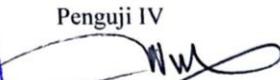
Penguji II


Mokhamad Iklil Mustofa, M.Kom
NIP. 198808072019031010

Penguji III


Dr. Khotibul Umam, M.Kom
NIP. 197908272011011007

Penguji IV


Adzhan Arwani Mahfudh, M.Kom
NIP. 199107032019031006

Pembimbing I


Wenty Dwi Yuniarti, S.Pd., M.Kom
NIP. 197706222006042005

Pembimbing II,


Mokhamad Iklil Mustofa, M.Kom
NIP. 198808072019031010



NOTA PEMBIMBING

Semarang, 26 Maret 2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

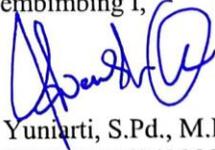
Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah dengan:

Judul : Analisis Sentimen Kebijakan
Program Pema'hadan Tahun 2023
Uin Walisongo Semarang Dengan
Metode *Naive Bayes Classifier*
Nama : **Viorella Amanda Putri**
NIM : 2008096025
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. Wr. Wb.

Pembimbing I,



Wenty Dwi Yuniarti, S.Pd., M.Kom
NIP. 197706222006042005

NOTA PEMBIMBING

Semarang, 26 Maret 2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

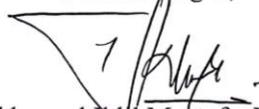
Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah dengan:

Judul : Analisis Sentimen Kebijakan
Program Pema'hadan Tahun 2023
Uin Walisongo Semarang Dengan
Metode *Naive Bayes Classifier*
Nama : **Viorella Amanda Putri**
NIM : 2008096025
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. Wr. Wb.

Pembimbing II,



Mokhamad Ikul Mustofa, M.Kom
NIP. 198808072019031010

ABSTRAK

Program pema'hadan bagi mahasiswa UIN merupakan program mandatori dari Kementerian Agama melalui Keputusan Direktur Jenderal Pendidikan Islam Nomor 7272 Tahun 2019 tentang Pedoman Implementasi Moderasi Beragama pada Pendidikan Islam. Sejak diumumkannya pada tanggal 17 April 2023 kebijakan tersebut menuai beragam komentar baik dukungan maupun tidak mendukung yang disampaikan melalui kolom komentar Instagram @uinwalisongosemarang.

Komentar yang masuk dari Instagram tersebut dapat dimanfaatkan untuk dilakukan analisis sentimen. Pada penelitian ini peneliti melakukan analisis sentimen yang diklasifikasikan menjadi positif dan negatif dari opini publik terhadap Kebijakan Program Pema'hadan Tahun 2023 UIN Walisongo Semarang dengan algoritma *Naïve Bayes Classifier* dan dengan metode CRISP-DM serta dilakukan perbandingan teknik *oversampling* antara SMOTE dengan ADASYN.

Hasil analisis sentimen terhadap data komentar Instagram tersebut yaitu dengan algoritma *Naïve Bayes* yang menggunakan SMOTE mendapatkan nilai akurasi sebesar 70%, *precision* sebesar 73%, *recall* sebesar 70% serta *f1 score* sebesar 71%. Sedangkan dengan algoritma *Naïve Bayes* yang menggunakan ADASYN mendapatkan nilai akurasi sebesar 68%, presisi sebesar 72%, *recall* sebesar 68%, *f1-score* sebesar 68%. Sehingga diperoleh kesimpulan bahwa penggunaan SMOTE dapat meningkatkan performa *Naïve Bayes* lebih tinggi dibandingkan dengan ADASYN dalam analisis sentimen.

Kata Kunci : analisis sentimen, kebijakan, UIN Walisongo, *Naïve Bayes Classifier*

KATA PENGANTAR

Alhamdulillah, Puji Syukur atas kehadiran Allah SWT atas segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Analisis Sentimen Kebijakan Program Pema’hadan Tahun 2023 UIN Walisongo Semarang dengan Metode *Naïve Bayes Classifier***” dengan baik. Adapun tujuan dibuat skripsi ini sebagai salah satu bentuk syarat kelulusan pada Program Sarjana (S1) Program Studi Teknologi Informasi di Universitas Islam Negeri Walisongo Semarang. Di sisi lain, penulis juga bertujuan untuk memberikan pengetahuan kepada pembaca.

Pada kesempatan ini penulis ingin mengucapkan banyak-banyak terimakasih kepada seluruh pihak yang telah memberi dukungan dan bantuan dari pelaksanaan skripsi hingga penyelesaian skripsi ini. Penulis mengakui bahwa apabila tanpa dibimbing, diberi arahan, dibina, dan tanpa diberikan motivasi dari seluruh pihak, maka penulisan skripsi ini tidak akan berjalan dengan baik. Maka dari itu penulis mengucapkan terimakasih yang tak terhingga kepada :

1. Bapak Prof. Dr. H. Nizar, M.Ag, selaku Rektor Universitas Islam Negeri Walisongo Semarang.
2. Bapak Prof. Dr. H. Musahadi, M.Ag, selaku Dekan Fakultas Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.

3. Bapak Dr. Khotibul Umam, S.T., M.Kom, selaku Ketua Program Studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
4. Ibu Wenty Dwi Yuniarti, S.Pd., M.Kom, dan Bapak Mokhammad Iklil Mustofa, M.Kom, selaku Dosen Pembimbing skripsi saya yang selalu memberikan dukungan, arahan, bimbingan serta motivasi dalam pelaksanaan skripsi hingga pembuatan skripsi ini.
5. Bapak Suwarno, S.Ag., M.Pd, yang telah membantu dalam proses pelabelan data dan validator.
6. Staff, karyawan dan dosen di lingkungan Universitas Islam Negeri Walisongo Semarang.
7. Orang tua tercinta dan keluarga yang selalu menemani dalam membantu penulis dan selalu mendo'akan serta memberikan dukungan baik kepada penulis.
8. M. Zain Alfyan Muis dan keluarga yang telah menemani, memberikan bantuan, dan dukungan baik kepada penulis.
9. Teman seperjuangan dari Ngawi dan teman di Kos J 9 yang telah menemani, membagikan pengalaman, dan semangat kepada penulis.
10. Teman-teman Teknologi Informasi yang selalu memberi dukungan.

11. Semua pihak yang tidak bisa penulis sebutkan satu persatu yang terlibat dalam pembuatan skripsi ini sehingga dapat terselesaikan dengan baik.

Dalam pelaksanaan dan penyusunan skripsi, penulis menyadari bahwa tentunya masih jauh dari kata sempurna dan masih banyak kekurangan. Untuk itu, penulis sangat mengharapkan kritik serta saran yang membangun demi kesempurnaan penulisan skripsi ini, dan semoga skripsi ini dapat bermanfaat untuk semua pihak. *Aamiin Yaarabbal 'Alamin.*

Semarang, 26 Maret 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN.....	ii
LEMBAR PENGESAHAN.....	iii
NOTA PEMBIMBING	iv
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN	1
A. Latar Belakang.....	1
B. Rumusan Masalah	6
C. Tujuan Penelitian.....	7
D. Batasan Masalah.....	7
E. Manfaat Penelitian.....	8
BAB II LANDASAN PUSTAKA.....	10
A. Kajian Teori.....	10
1. <i>Text Mining</i>	10
2. <i>Text Preprocessing</i>	11
3. Instagram	14
4. Analisis Sentimen	15
5. TF-IDF.....	17
6. <i>Split Validation Data</i>	19

7. <i>SMOTE</i>	20
8. <i>Adaptive Synthetic (ADASYN)</i>	21
9. <i>Naïve Bayes Classifier</i>	22
10. Evaluasi.....	24
B. Kajian Pustaka.....	27
BAB III METODE PENELITIAN	30
A. Metode Pengumpulan Data	30
1. Studi Literatur	30
2. Studi Lapangan	30
B. Perangkat Penelitian.....	31
C. Alur Pengerjaan Penelitian.....	32
BAB IV HASIL DAN PEMBAHASAN.....	46
A. <i>Bussines Understanding</i>	46
B. <i>Data Understanding</i>	48
C. <i>Data Preparation</i>	52
1. <i>Preprocessing</i>	52
a) <i>Case Folding</i>	52
b) <i>Cleansing</i>	53
c) <i>Normalization</i>	55
d) <i>Tokrnization</i>	57
e) <i>Stopword Removal</i>	58
f) <i>Stemming</i>	60
g) <i>Remove Comment</i>	61
2. <i>Processing</i>	63
a) <i>Split Validation</i>	64

b) Pembobotan TF-IDF.....	64
c) <i>Resampling</i>	66
D. <i>Modeling</i>	70
E. <i>Evaluation</i>	72
F. <i>Deployment</i>	81
BAB V KESIMPULAN	85
A. Kesimpulan	85
B. Saran.....	86
DAFTAR PUSTAKA.....	88
LAMPIRAN	93

DAFTAR TABEL

Tabel 2.1 <i>Confusion Matrix 2x2</i>	25
Tabel 2.2 Penelitian Terdahulu	27
Tabel 3.1 Kebutuhan Perangkat Keras	31
Tabel 3.2 Kebutuhan Perangkat Lunak	31
Tabel 3.3 Contoh Hasil Pengambilan Data Instagram	35
Tabel 3.4 Contoh Hasil Pelabelan	36
Tabel 3.5 Contoh Hasil <i>Case Folding</i>	37
Tabel 3.6 Contoh Hasil <i>Cleansing</i>	38
Tabel 3.7 Contoh Hasil <i>Normalization</i>	39
Tabel 3.8 Contoh Hasil <i>Tokenization</i>	39
Tabel 3.9 Contoh Hasil <i>Stopword Removal</i>	40
Tabel 3.10 Contoh Hasil <i>Stemming</i>	41

DAFTAR GAMBAR

Gambar 3.1 Alur Metode CRISP_DM	32
Gambar 3.2 Diagram Alur Penelitian.....	33
Gambar 4. 1 Tampilan IGCommentExporter	47
Gambar 4. 2 Contoh Data Hasil Ekspor	47
Gambar 4. 3 Tampilan Aplikasi Jupyter Notebook.....	48
Gambar 4. 4 Contoh Data Berlabel	49
Gambar 4. 5 Contoh Kamus Perbaikan	50
Gambar 4. 6 Kode Pemanggilan <i>Library</i>	51
Gambar 4. 7 Kode Penerapan Proses <i>Case Folding</i>	53
Gambar 4. 8 Hasil Penerapan <i>Case Folding</i>	53
Gambar 4. 9 Kode Pemanggilan <i>Library Re Dan Emoji</i>	54
Gambar 4. 10 Kode Penerapan Proses <i>Cleansing</i>	54
Gambar 4. 11 Hasil Penerapan <i>Cleansing</i>	55
Gambar 4. 12 Kode Pemanggilan Kamus Perbaikan Kata 1	56
Gambar 4. 13 Kode Pemanggilan Kamus Perbaikan Kata 2	56
Gambar 4. 14 Kode Penerapan <i>Normalization 1</i>	56
Gambar 4. 15 Kode Penerapan <i>Normalization 2</i>	57
Gambar 4. 16 Hasil Penerapan <i>Normalzation</i>	57
Gambar 4. 17 Kode Penerapan <i>Tokenization</i>	58
Gambar 4. 18 Hasil Implementasi <i>Tokenization</i>	58
Gambar 4. 19 Pemanggilan Modul <i>Stopwords</i>	59
Gambar 4. 20 Kode Penerapan <i>Stopword Removal</i>	59
Gambar 4. 21 Hasil Penerapan <i>Stopword Removal</i>	60
Gambar 4. 22 Kode Pemanggilan <i>Modul Stemmer Factory</i> .	60

Gambar 4. 23 Kode Penerapan <i>Stemming</i>	61
Gambar 4. 24 Hasil Penerapan <i>Stemming</i>	61
Gambar 4. 25 Kode Penghapusan Komentar Kosong	62
Gambar 4.26 Hasil Kode Penghapusan Komentar Kosong..	62
Gambar 4. 27 Hasil Total Sentimen Pada Data Komentar ...	63
Gambar 4. 28 Kode Pembagian Data <i>Training</i> Dan <i>Testing</i>	64
Gambar 4. 29 Hasil Penerapan TF-IDF.....	65
Gambar 4. 30 Hasil Proses TF-IDF.....	66
Gambar 4. 31 Kode Penerapan <i>Oversampling</i> SMOTE.....	67
Gambar 4. 32 Kode Penerapan <i>Oversampling</i> ADASYN....	68
Gambar 4. 33 Hasil Implementasi <i>Oversampling</i> SMOTE..	68
Gambar 4. 34 Hasil Implementasi <i>Oversampling</i> ADASYN	69
Gambar 4. 35 Perbandingan Data Sebelum Dan Setelah <i>Oversampling</i> SMOTE.....	70
Gambar 4. 36 Perbandingan Data Sebelum Dan Setelah <i>Oversampling</i> ADASYN.....	70
Gambar 4. 37 Kode Pemanggilan Modul <i>Library Sklearn</i> ...	71
Gambar 4. 38 Kode Proses Pengklasifikasian <i>Naive Bayes</i> Dengan SMOTE.....	71
Gambar 4. 39 Kode Proses Pengklasifikasian <i>Naive Bayes</i> Dengan ADASYN.....	72
Gambar 4. 40 <i>Confusion Matrix</i> Data SMOTE.....	73
Gambar 4. 41 Nilai Performa Tiap Kelas Dari Pemodelan NBC Dengan SMOTE	75
Gambar 4. 42 <i>Confusion Matrix</i> Data ADASYN.....	77

Gambar 4. 43 Nilai Performa Tiap Kelas Dari Pemodelan NBC Dengan ADASYN	80
Gambar 4.44 Persentase Pada Sentimen	81
Gambar 4.45 <i>Wordcloud</i> Data Dari Instagram.....	82
Gambar 4.46 <i>Wordcloud</i> Pada Sentimen Positif	83
Gambar 4.47 <i>Wordcloud</i> Pada Sentimen Negatif.....	84

DAFTAR LAMPIRAN

Lampiran 1. Contoh Hasil Ekspor Data Instagram	93
Lampiran 2. Contoh File Yang Sudah Diberi Label	94
Lampiran 3. Contoh File “Kamus bahasa tidak baku.csv”...	95
Lampiran 4. Contoh File “kata tidak baku 2.csv”	96
Lampiran 5. Contoh File “Stopwords.txt”	97
Lampiran 6. Dokumen Hasil Klasifikasi NBC Dengan SMOTE	98
Lampiran 7. Dokumen Hasil Klasifikasi NBC Dengan ADASYN	101
Lampiran 8. <i>Source Code</i>	104
Lampiran 9. Contoh Data Hasil <i>Oversampling</i> SMOTE.....	109
Lampiran 10. Contoh Data Hasil <i>Oversampling</i> ADASYN.	110

BAB I

PENDAHULUAN

A. Latar Belakang

Teknologi informasi dan komunikasi khususnya pada internet berkembang begitu cepat setiap tahunnya, sehingga pengguna internet juga semakin banyak mengikuti perkembangannya. Pengguna internet di Indonesia menjadi salah satu pengguna internet aktif terbanyak di dunia dan kemungkinan akan terus naik setiap tahunnya (Dedy et al., 2023). Menurut data APJII menyebutkan pengguna internet di Indonesia mencapai 215,63 juta orang pada 2022-2023. Jumlah tersebut meningkat 2,67 persen dibandingkan pada periode sebelumnya yakni sebanyak 210,03 juta pengguna. Hal ini membawa perubahan dalam penyebaran suatu informasi yang mana dengan adanya internet menjadikan penyebaran informasi menjadi mudah disampaikan dan pengguna internet mudah dalam mendapatkan informasi.

Sosial media menjadi *platform digital* yang banyak digunakan sebagai media yang dapat menjangkau banyak pengguna internet dalam penyebaran informasi. Sosial media adalah sebuah *platform digital* yang memudahkan para penggunanya untuk bertukar informasi dan mengekspresikan diri baik berupa foto, video, maupun berupa tulisan. Salah satu sosial media yang populer digunakan di Indonesia yaitu Instagram.

Pengguna Instagram di Indonesia memiliki jumlah yang besar yakni 104,8 juta pengguna pada bulan Oktober 2023 yang menjadikan Indonesia menduduki peringkat ke-4 dunia menurut laporan *We Are Social*. Kemungkinan data tersebut akan terus meningkat sehingga menjadikan Instagram sebagai media yang cocok untuk melakukan *branding*, publikasi, dan promosi yang efektif. Potensi tersebut patut untuk dimanfaatkan bagi individu maupun suatu instansi sebagai media informasi yang intensif menyebarkan informasi karena dapat menjangkau banyak pengguna internet terutama Instagram (Febilianingtyas & Febriana, 2022). Melalui Instagram tersebut pengguna dapat melakukan interaksi dengan melakukan *like* maupun memberikan komentar pada suatu unggahan.

Salah satu instansi yang turut memanfaatkan potensi Instagram tersebut yaitu Universitas Islam Negeri Walisongo yang merupakan salah satu perguruan tinggi negeri di Semarang. Universitas Islam Negeri Walisongo Semarang atau UIN Walisongo Semarang memanfaatkan Instagram sebagai media informasi dan *branding* untuk perguruan tinggi. Akun Instagram UIN Walisongo Semarang yakni @uinwalisongosemarang saat ini, September 2023, memiliki jumlah pengikut sebesar 69.2 ribu pengikut. Beragam informasi yang diunggah pada akun Instagram tersebut seperti pengumuman, capaian prestasi, bahkan informasi terkait kebijakan kampus.

Salah satu informasi yang diunggah pada @uinwalisongosemarang yakni informasi terkait Kebijakan Program Pema'hadan Tahun 2023 yang diunggah pada tanggal 17 April 2023. Unggahan tersebut memuat informasi Kebijakan Program Pema'hadan Tahun 2023 berdasarkan surat keputusan rektor dengan nomor 1901/Un.10.0/R1/DA.00.01/04/2023 yang ditandatangani pada tanggal 11 April 2023 oleh Wakil Rektor Bidang Akademik dan Kelembagaan yakni Prof. Dr. M. Mukhsin Jamil, M.Ag. Unggahan tersebut mengundang beragam opini yang disampaikan pada kolom komentarnya. Melalui Instagram ini, pengguna lain dapat berinteraksi, beropini, dan memberikan penilaian secara terbuka kepada UIN Walisongo Semarang dengan cara memberikan komentar langsung pada akun tersebut.

Dalam Kebijakan Program Pema'hadan tersebut, salah satunya disebutkan bahwa semua mahasiswa baru UIN Walisongo Semarang Tahun 2023 wajib mengikuti Program Pema'hadan. Program Pema'hadan bagi mahasiswa UIN merupakan Program Mandatori dari Kementerian Agama melalui Keputusan Direktur Jenderal Pendidikan Islam Nomor 7272 Tahun 2019 tentang Pedoman Implementasi Moderasi Beragama pada Pendidikan Islam. Kebijakan tersebut menuai beragam komentar baik yang menyatakan dukungan maupun tidak, pernyataan tersebut disampaikan melalui kolom komentar Instagram @uinwalisongosemarang.

Komentar yang masuk dari Instagram tersebut dapat dimanfaatkan sebagai parameter kunci untuk memberikan gambaran penilaian masyarakat terhadap kinerja atau kebijakan dari suatu lembaga dengan menggunakan analisis sentimen. Menurut Azizah (2023), analisis sentimen merupakan cabang ilmu dari *text mining* yang dipakai untuk mengekstrak dan mengolah data berbentuk teks. Analisis sentimen merupakan bidang ilmu yang mempelajari cara untuk melakukan analisis terhadap opini, pendapat, dan penilaian dari suatu pihak tertentu yang dapat berupa layanan, produk, pihak, dan isu tertentu (Husada, H.C., et al., 2021). Analisis sentimen penting untuk dilakukan karena untuk mengetahui kemungkinan kecenderungan penilaian masyarakat apakah memberikan respon yang positif, atau negatif. Hasil analisis tersebut dapat digunakan sebagai acuan dalam melakukan strategi promosi, pengambilan keputusan, dan peningkatan layanan universitas, serta sebagai evaluasi.

Evaluasi atau muhasabah perlu dilakukan sesuai dengan firman Allah dalam Al-Qur'an surah Al-Hasyr ayat 18 sebagai berikut:

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَلْتَنْظُرْ نَفْسٌ مَّا قَدَّمَتْ لِإِعَادَةٍ وَأَتَّقُوا اللَّهَ إِنَّ اللَّهَ خَبِيرٌ بِمَا تَعْمَلُونَ

Artinya : *“Wahai orang-orang yang beriman, bertakwalah kepada Allah. Hendaklah setiap orang memperhatikan apa yang telah diperbuatnya untuk hari esok. Bertakwalah kepada Allah.*

Sungguh, Allah Maha Mengetahui apa yang kamu kerjakan” (QS Al-Hasyr ayat 18).

Di dalam Tafsir Al- Misbah menjelaskan bahwa ayat tersebut merupakan suatu ajakan untuk senantiasa bertakwa kepada Allah serta mencermati perbuatan kita, dan membetulkan amal yang kurang baik di masa lalu dan bernilainya muhasabah ataupun penilaian diri ditekankan dalam rangka membetulkan diri serta membetulkan mutu hidup (Ayu Purnamasari S, Muhammad Amin, 2022). Dari penjelasan ayat tersebut diharapkan bahwa analisis sentimen ini dapat digunakan sebagai bahan evaluasi untuk meningkatkan mutu layanan universitas.

Metode yang dapat digunakan dalam melakukan analisis sentimen salah satunya yakni dengan menggunakan metode *Naïve Bayes*. Metode *Naïve Bayes Classifier* dipilih karena kesederhanaan, efektif dan kompatibilitas menangkap visualisasi model data (Mustofa & Mahfudh, 2019). Metode *Naïve Bayes* juga merupakan sebuah algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Susianti et al., 2020).

Sebagai upaya untuk meningkatkan performa yang diberikan algoritma *Naïve Bayes*, peneliti melakukan penyeimbangan data dengan menerapkan teknik *oversampling*. Menurut Yerik Afrianto Singgalen (2023), pengolahan algoritma yang tidak mempertimbangkan ketidakseimbangan data cenderung menitikberatkan kelas mayor dan bukan kelas minor,

oleh sebab itu diperlukan *oversampling* untuk memperbanyak pengamatan secara acak dengan menambah jumlah data kelas minor (data buatan) agar setara dengan kelas mayor.

Penelitian ini menggunakan dua teknik *oversampling* yakni teknik SMOTE (*Synthetic Minority Oversampling Technique*) dan ADASYN (*Adaptive Synthetic*), untuk membandingkan teknik mana yang memberikan performa lebih baik apabila disandingkan dengan algoritma *Naïve Bayes Classifier*. Menurut penelitian yang dilakukan oleh Fajar Muharram dan Kana Saputra S., (2023), performa yang diberikan oleh algoritma *Naïve Bayes* dengan SMOTE memiliki tingkat akurasi lebih baik yaitu sebesar 78% dibandingkan tanpa menggunakan SMOTE sebesar 70%. Sedangkan penelitian yang dilakukan oleh Luthfiyyah et al., (2024), performa yang diberikan oleh algoritma *Naïve Bayes* dengan ADASYN memiliki akurasi lebih tinggi yaitu sebesar 89% dibandingkan tanpa menggunakan ADASYN yaitu 87%. Maka dari itu, perlu dilakukan perbandingan antara *Naïve Bayes* dengan SMOTE dan *Naïve Bayes* dengan ADASYN manakah yang memiliki performa lebih baik dalam melakukan analisis sentimen Kebijakan Program Pema'hadan Tahum 2023 UIN Walisongo Semarang.

B. Rumusan Masalah

Berdasarkan uraian latar belakang diatas, rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana menerapkan algoritma *Naïve Bayes Classifier* dengan SMOTE dan algoritma *Naïve Bayes Classifier* dengan ADASYN untuk analisis sentimen Kebijakan Program Pema'hadan Tahun 2023 ?
2. Bagaimana performa yang diberikan dari perbandingan antara algoritma *Naïve Bayes Classifier* dengan SMOTE dan *Naïve Bayes Classifier* dengan ADASYN pada analisis sentimen Kebijakan Program Pema'hadan Tahun 2023 ?

C. Tujuan Penelitian

Tujuan dari penelitian ini berdasarkan rumusan masalah diatas adalah sebagai berikut :

1. Menerapkan algoritma *Naïve Bayes Classifier* dengan SMOTE dan algoritma *Naïve Bayes Classifier* dengan ADASYN untuk analisis sentimen Kebijakan Program Pema'hadan Tahun 2023.
2. Mengetahui performa yang diberikan oleh algoritma *Naïve Bayes Classifier* dengan SMOTE dan algoritma *Naïve Bayes Classifier* dengan ADASYN untuk analisis sentimen Kebijakan Program Pema'hadan Tahun 2023.

D. Batasan Masalah

Batasan masalah perlu diterapkan pada penelitian ini, agar penelitian dapat dilakukan secara objektif dan jelas. Batasan tersebut adalah sebagai berikut :

1. Data yang digunakan adalah data dari komentar salah satu unggahan pada media sosial Instagram dari akun @uinwalisongosemarang tanggal 17 April 2023 terkait pengumuman Kebijakan Program Pema'hadan Tahun 2023, sebanyak 565 komentar.
2. Peneliti melakukan perbandingan analisis sentimen yaitu antara algoritma *Naïve Bayes Classifier* dengan teknik *oversampling* SMOTE dan algoritma *Naïve Bayes Classifier* dengan teknik *oversampling* ADASYN.
3. Komentar pada Instagram akan diklasifikasikan menjadi dua sentimen yaitu sentimen positif dan negatif.
4. Data komentar terkait Kebijakan Program Pema'hadan ini diambil sejak tanggal 17 April - 31 Oktober 2023.

E. Manfaat Penelitian

Manfaat yang diharapkan dari hasil analisis sentimen ini adalah

1. Manfaat teoritis :
 - a. Membantu untuk mengklasifikasi komentar menjadi tanggapan positif dan negatif.
 - b. Mengetahui tingkat performa pada algoritma *Naïve Bayes Classifier* dalam melakukan klasifikasi pada Kebijakan Program Pema'hadan Tahun 2023 UIN Walisongo Semarang.

c. Sebagai pijakan, bahan referensi dan pengembangan pada penelitian-penelitian selanjutnya yang berhubungan dengan penelitian sentimen analisis.

2. Manfaat praktis :

a. Bagi Masyarakat

Dapat mengetahui jumlah respon/tanggapan terhadap kebijakan universitas mengenai program pema'hadan apakah lebih cenderung positif atau negatif.

b. Bagi Universitas

- Dapat menjadi bahan pertimbangan universitas dengan mengetahui sentimen masyarakat terkait Kebijakan Program Pema'hadan Tahun 2023 UIN Walisongo Semarang.
- Dapat menganalisa terhadap hasil Kebijakan Program Pema'hadan Tahun 2023 dan meminimalisirkan masalah atau solusi dimasa yang akan datang berdasarkan hasil sentimen tersebut.

BAB II

LANDASAN PUSTAKA

A. Kajian Teori

1. *Text Mining*

Text mining merupakan proses memperoleh suatu informasi dari dokumen berbentuk teks yang menjadi sumbernya. Menurut Husada, H.C., et al., (2021) *text mining* merupakan alat yang digunakan untuk proses pengambilan keputusan dengan memanfaatkan data teks dengan jumlah banyak, dan *text mining* sering digunakan untuk menganalisis informasi. Afdhal et al., (2022) mendefinisikan *text mining* sebagai variasi dari data mining yang berupaya untuk memperoleh pola yang menarik dari sekumpulan data tekstual yang berjumlah besar. *Text mining* yaitu proses menambang data yang berupa teks, dimana sumber yang yang dihasilkan biasanya didapatkan dari suatu dokumen (Harpizon et al., 2022).

Menurut Husada, H.C., et al., (2021) tujuan *text mining* yaitu untuk memperoleh pola guna tujuan yang spesifik, menurut Zhafira et al., (2021) pola tersebut berupa pola statistik. Sedangkan menurut (Susianti et al., 2020) tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan

dokumen. Harpizon et al., (2022) menyebutkan pula tujuan dari *text mining* ini yaitu untuk mencari kata-kata dan mengekstrak informasi yang berguna dari sumber data yang dapat mewakili isi dari suatu dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Text mining juga memiliki tugas khusus diantaranya yaitu dapat digunakan untuk melakukan klasifikasi sentimen, *text clustering* (pengelompokan teks), *information extraction* dan *information retrieval*. Dokumen teks yang digunakan biasanya masih terdapat *noise*, sehingga perlu dilakukan *text preprocessing* terlebih dahulu.

2. *Text Preprocessing*

Dokumen teks yang digunakan dalam teks mining biasanya masih terdapat *noise*, untuk membersihkannya perlu dilakukan *text preprocessing* terlebih dahulu. *Text preprocessing* merupakan proses untuk mempersiapkan data mentah sebelum dilakukan proses lain (Susianti et al., 2020). *Text preprocessing* atau pengolahan data awal dilakukan untuk menyederhanakan atau menghilangkan bagian-bagian yang tidak diperlukan (*noise*) untuk proses klasifikasi agar menjadi lebih efisien dan akurat (Dedy et al., 2023). Khairunnisa et al., (2021) menyatakan bahwa

text preprocessing digunakan untuk transformasi tata data menjadi lebih terstruktur sehingga menghasilkan informasi teks dengan kualitas yang baik dan siap digunakan pada proses selanjutnya.

Dalam penelitian ini langkah *text preprocessing* terdiri atas beberapa tahapan sebagai berikut:

a. *Case Folding*

Case folding merupakan proses perubahan karakter dari huruf kapital pada data yang ditransformasi menjadi huruf kecil (Zhafira et al., 2021). Tujuan dari *case folding* ini adalah untuk menyamaratakan bentuk teks agar kata-kata yang sama tidak terdeteksi berbeda hanya karena perbedaan terdapat huruf kapital.

b. *Cleansing*

Cleansing merupakan proses pemilihan kata yang akan digunakan dengan membuang *noise* diantaranya tanda baca, angka, emoji, spasi ganda, baris *enter*, alamat *website*, karakter *HTML*, simbol *emoticon*, *hashtag* (#), nama pengguna (@*username*), dan *link URL* (Afdhal et al., 2022).

c. *Remove Duplicate*

Pada langkah *remove duplicate*, data komentar ganda atau data yang memiliki isi sama akan dilakukan penghapusan.

d. *Normalization*

Normalization merupakan langkah menyesuaikan kata yang salah pengejaan menjadi kata yang sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Tujuan dari langkah ini yaitu diharapkan peneliti dapat terbantu untuk mendapatkan ejaan kata yang tepat (Azizah, 2023).

e. *Stopword Removal*

Stopword removal merupakan tahapan penyaringan kata-kata penting dari data yang didapat dengan menghilangkan kata yang tidak digunakan sesuai dengan *list stopwords*. Kata yang ada di dalam *list stopwords* akan dihapus karena dianggap tidak memengaruhi hasil analisis sentimen (Zhafira et al., 2021).

f. *Tokenization*

Tokenization merupakan proses pemisahan kata-kata dari kalimat penyusunnya pada teks menjadi bagian tertentu yang setelah itu disebut dengan token (Tuhuteru et al., 2018).

g. *Stemming*

Stemming merupakan proses pemetaan berbagai variasi dari morfologi kata yang dikembalikan ke bentuk dasarnya (Indraloka & Santosa, 2017 dalam Zhafira et al., 2021). Proses *stemming* juga

mengubah kata yang perulangan menjadi satu kata dasar.

3. Instagram

Instagram merupakan salah satu media sosial yang populer di Indonesia. Pengguna Instagram di Indonesia memiliki jumlah yang besar yakni 89,15 juta yang menjadikan Indonesia menduduki peringkat ke-4 dunia menurut CNBC Indonesia. Kemungkinan data tersebut akan terus meningkat sehingga menjadikan Instagram sebagai media yang cocok untuk melakukan *branding*, promosi, dan penyampaian informasi yang efektif. Potensi tersebut patut untuk dimanfaatkan bagi individu maupun suatu instansi sebagai media informasi yang intensif menyebarkan informasi karena dapat menjangkau khalayak yang beragam (Febilianingtyas & Febriana, n.d.).

Instagram adalah *platform* sosial media yang mana penggunaanya dapat melakukan beragam interaksi bagi sesama pengguna lainnya. Ragam interaksi tersebut diantaranya yaitu pengguna dapat mengunggah foto dan video, menyimpan unggahan dari pengguna lain, memungkinkan penggunaanya untuk memberikan komentar dan meninggalkan tanda suka pada suatu unggahan (Flores et al., 2020).

Kelebihan dari Instagram tersebut dapat dimanfaatkan datanya untuk berbagai keperluan salah satunya adalah untuk melakukan analisis sentimen.

4. Analisis Sentimen

Menurut Permana (2023) analisis sentimen adalah proses mengidentifikasi polaritas teks dalam dokumen atau jenis dokumen lainnya sehingga kategori dapat ditandai sebagai sentimen positif, negatif, atau netral. Analisis sentimen atau bisa disebut juga dengan *opinion mining* merupakan penerapan dari ilmu komputasi linguistik, *text analysis*, dan *natural language processing* yang digunakan untuk mengekstrak subjektivitas dari sebuah sumber teks. Tujuan dari analisis sentimen sendiri yaitu digunakan untuk memberikan pernyataan suka atau tidak suka terhadap suatu hal seperti layanan, kebijakan, produk atau suatu peristiwa melalui sentimen positif dan negatif yang mana pernyataan tersebut dapat digunakan sebagai bahan pertimbangan untuk proses pengambilan keputusan (Husada, H.C., et al., 2021).

Menurut Azizah, (2023) analisis sentimen adalah cabang ilmu dari *data mining*, yang mana untuk menjadikan informasi dapat dipahami dengan baik, data diproses dengan beberapa cara yang dimulai dengan cara memahami teks, mengekstrak, dan

mengolah data yang berbentuk kalimat secara otomatis.

Dilihat dari asal datanya analisis sentimen dibagi menjadi 2 kelompok yaitu :

- a. *coarse-grained sentimen analysis* yakni sentimen analisis yang dilakukan pada level dokumen, secara garis besar fokus utama dari sentimen analisis jenis ini adalah menganggap seluruh isi dokumen sebagai sebuah sentimen positif dan sentimen negatif.
- b. *Fined-grained sentimen analysis* adalah sentimen analisis yang dilakukan pada level kalimat, fokus utamanya adalah menentukan sentimen pada setiap kalimat.

Sentimen analisis dapat dibagi menjadi beberapa kelas sentimen diantaranya kelas sentimen bersifat positif dan negatif.

- a. Sentimen positif: Menurut Kamus Besar Bahasa Indonesia (KBBI) sentimen positif merupakan reaksi atau sikap yang meningkatkan nilai seseorang atau sesuatu (Ardiani et al., 2020).
- b. Sentimen negatif: Menurut Kamus Besar Bahasa Indonesia (KBBI) sentimen negatif merupakan reaksi atau sikap yang

menurunkan nilai seseorang atau sesuatu, jadi kalimat bersentimen negatif akan menyebabkan penyurutan nilai pandang terhadap sesuatu, sehingga membentuk *trend down* (Ardiani et al., 2020).

5. TF-IDF

Metode *Term Frequency Inverse Document Frequency* (TF-IDF) merupakan metode pembobotan yang paling umum digunakan, yang mana pembobotan tersebut merupakan gabungan dari *term frequency* dan *inverse document frequency* (Susianti et al., 2020). TF-IDF digunakan untuk perhitungan bobot suatu kata dalam dokumen. Menurut Zhafira et al., (2021) TF-IDF merupakan penentuan seberapa *term* mewakili konten dalam dokumen dengan memberi bobot ke masing-masing kata yang terkandung di dalamnya (Karmayasa & Mahendra, 2010 dalam Zhafira et al., 2021).

Term Frequency (TF) sendiri fungsinya untuk melakukan pengukuran seberapa sering sebuah kata muncul pada suatu dokumen. Sedangkan *Inverse Document Frequency* (IDF) berfungsi untuk mengukur seberapa penting kata dalam suatu dokumen dengan melakukan pembobotan kata yang mana bobot yang paling kecil diberikan pada kata dengan

frekuensi kemunculan terbanyak, dan memberikan bobot terbesar bagi kata yang frekuensi kemunculannya paling jarang. Sehingga dapat disimpulkan bahwa besarnya jumlah kata yang muncul berbanding lurus dengan pembobotan yang diberikan (Husada, H.C., et al., 2021). Hal demikian terjadi karena metode pembobotan TF-IDF menganggap bahwa semakin sedikit tingkat frekuensi kemunculan kata pada sebuah dokumen maka kata tersebut akan dianggap semakin unik atau penting (Flores et al., 2020).

Nilai TF dan IDF didapat dari perhitungan Persamaan 2.1 dan 2.2 sebagai berikut :

$$\text{TF}(t,d) = \frac{\text{Jumlah kata } t \text{ pada dokumen } d}{\text{Total kata pada dokumen } d} \quad (2.1)$$

$$\text{IDF}(t) = \log \frac{\text{Total dokumen}}{\text{Jumlah dokumen yang mengandung kata } t} \quad (2.2)$$

Pada persamaan tersebut d merupakan dokumen sedangkan t merupakan kata. Nilai TF-IDF didapat dari perkalian nilai TF dan nilai IDF sehingga terbentuk persamaan 2.3 :

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t) \quad (2.3)$$

6. *Split Validation Data*

Sebelum dilakukan langkah pengklasifikasian, data yang telah dilakukan pembobotan dipisah menjadi dua data yang disebut dengan proses *split validation*. *Split validation* adalah teknik validasi yang membagi data menjadi dua bagian secara acak, bagian pertama digunakan sebagai data *training* dan bagian kedua digunakan sebagai data *testing*.

Data *training* adalah data yang akan dipakai dalam melakukan pembelajaran sedangkan data *testing* adalah data yang belum pernah dipakai sebagai pembelajaran dan akan berfungsi sebagai data pengujian. Memisahkan data menjadi *training* dan *testing set* dimaksudkan agar model yang diperoleh nantinya memiliki kemampuan generalisasi yang baik dalam melakukan klasifikasi data (Turmudi Zy et al., 2021).

Dengan menggunakan *split validation* akan dilakukan percobaan *training* berdasarkan *split ratio* yang telah ditentukan sebelumnya. Semakin banyak data latih yang digunakan cenderung meningkatkan nilai *accuracy* dari hasil klasifikasi (Zhafira et al., 2021)

7. *SMOTE (Synthetic Minority Oversampling Technique)*

SMOTE (Synthetic Minority Oversampling Technique) merupakan metode *oversampling* dimana berfungsi untuk memecahkan permasalahan ketidakseimbangan pada kelas. Teknik ini merupakan pendekatan untuk menyeimbangkan data sampel pada kelas yang memiliki ketidakseimbangan berlebihan (mayoritas) dengan fokus terhadap kelas minoritas, dengan tujuan meningkatkan kinerja dari klasifikasi dataset (Kurniawan et al., 2023). Model pembelajaran tradisional cenderung dilihat berpihak pada kelas mayoritas dan mengabaikan kinerja kelas minoritas. Maka dari itu penerapan SMOTE diperlukan untuk menangani ketidakseimbangan dataset.

Pertama dilakukan pendekatan pengklasifikasian dari sekelompok data yang tidak seimbang. Dimana kategori klasifikasi tidak disajikan secara seimbang. Kedua, algoritma yang terdapat pada SMOTE akan mengurangi nilai selisih antara vektor dari fitur pada kelas minoritas dan nilai *nearest neighbor* dari kelas minoritas lalu dikalikan dengan nilai acak antara 0 sampai 1. Kemudian, menambahkan hasil perhitungan dengan vektor fiturnya maka didapatkan hasil nilai

vektor yang baru. Berikut merupakan persamaan SMOTE :

$$X_{\text{new}} = (X_i + (X_i^{\wedge} - X_i) \times \delta) \quad (2.4)$$

Keterangan :

X_i = vektor dari fitur pada kelas minoritas

X_i^{\wedge} = *k-nearest neighbors* untuk X_i

δ = angka acak antara 0 sampai 1

8. *Adaptive Synthetic (ADASYN)*

ADASYN adalah salah satu teknik *oversampling* untuk meningkatkan kinerja model berdasarkan pada distribusi data. Metode ini bekerja dengan mengurangi bias yang disebabkan oleh ketidakseimbangan kelas, dan secara adaptif menggeser batas keputusan klasifikasi ke arah sampel yang sulit. Ide penting dari ADASYN adalah menggunakan pembobotan yang memperhatikan distribusi untuk sampel kelas minoritas menurut tingkat kesulitan untuk diklasifikasikan. Metode ini lebih banyak menghasilkan data sintesis untuk kelas minoritas yang lebih sulit dipelajari dibandingkan dengan contoh-contoh minoritas yang lebih mudah dipelajari. ADASYN dirancang untuk membuat

sampel sintetis di wilayah dengan kepadatan sampel minoritas rendah (Nurhopipah & Magnolia, 2022).

9. *Naïve Bayes Classifier*

Setelah dilakukan pembobotan dengan menggunakan TF-IDF, langkah selanjutnya adalah pengujian klasifikasi dengan metode *Naïve Bayes Classifier* (NBC). *Naïve Bayes Classifier* atau NBC merupakan proses pengklasifikasian probabilitas sederhana yang mengacu pada teori *Bayes*. Teori tersebut menyatakan bahwa kemungkinan terjadinya suatu peristiwa sama dengan probabilitas intrinsik (dihitung dari data yang tersedia sekarang) dikalikan probabilitas bahwa hal serupa akan terjadi lagi di masa depan (berdasarkan pengetahuan yang terjadinya di masa lalu) (Mustofa & Mahfudh, 2019). Tujuan dari pengujian klasifikasi ini yaitu untuk menghitung nilai probabilitasnya (Ningtyas et al., 2023).

Menurut Ningtyas et al., (2023) keuntungan dari menggunakan *Naïve Bayes Classifier* sebagai metode pengujian klasifikasinya adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*Training Data*) yang kecil untuk menentukan estimasi parameter. Selain itu menurut Chely Aulia Misrun et al., (2023) kelebihan algoritma *Naïve Bayes Classifier* yaitu mudah di implementasikan dan mempunyai

tingkat akurasi yang tinggi. Peneliti lain juga menyebutkan bahwa *Naïve Bayes Classifier* merupakan metode dalam penambangan data yang populer dikarenakan kemudahan penggunaannya (Hall, 2006 dalam Zhafira et al., 2021).

Gagasan yang mendasari aturan *Bayes* yaitu, hasil dari hipotesis atau peristiwa (A) dapat diperkirakan berdasarkan pada beberapa *evidence* (X) yang diamati. Hal penting dalam *Bayes* adalah sebuah probabilitas awal/priori A atau P(A) adalah probabilitas dari suatu hipotesis sebelum bukti diamati. Sebuah probabilitas posterior A atau P(A|X) adalah probabilitas dari suatu hipotesis setelah bukti-bukti yang diamati ada (Yuniarti et al., 2020).

Rumus *Naïve Bayes Classifier* dapat dilihat pada persamaan 2.4 :

$$P(A|X) = \frac{P(X|A) \times P(A)}{P(X)} \quad (2.5)$$

Keterangan :

A = Data hipotesis X dari kelas yang spesifik

X = Data kelas yang tidak diketahui

- $P(A|X)$ = Probabilitas hipotesis A berdasarkan kondisi X (posterior)
- $P(X|A)$ = Probabilitas X berdasarkan kondisi tersebut (*Likelihood*)
- $P(A)$ = Peluang terjadinya A (*prior*)
- $P(X)$ = Peluang terjadinya X (*evidence*)

10. Evaluasi

Proses evaluasi merupakan proses yang dilakukan untuk menguji hasil dari klasifikasi dengan mengukur nilai performansi dari sistem yang telah dibuat untuk mengukur keakuratan perolehan hasil tes yang terbaik (Rahayu et al., 2022). Pada proses ini metode pengukuran yang digunakan adalah *confusion matrix (classification matrix or a contingency table)* (Susianti et al., 2020).

Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode kalsifikasi untuk mendeteksi kerusakan. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (Turmudi Zy et al., 2021). Berikut tabel *confusion matrix 2x2*:

Tabel 2.1 *Confusion Matrix 2X2*

<i>Predicted Class</i>	<i>True Class</i>	
	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	TP = <i>True Positive</i>	FN = <i>False Negative</i>
<i>Negative</i>	FP = <i>False Positive</i>	TN = <i>True Negative</i>

Pada pengukuran kinerja menggunakan *Confusion matrix*, terdapat 4 (empat) istilah representasi hasil proses klasifikasi sebagai berikut:

- a. TP (*true positive*) : Yaitu data yang diprediksi positif dan faktanya data itu positif (Sesuai).
- b. TN (*true negative*) : Yaitu data yang diprediksi negatif dan faktanya data itu negatif (Sesuai).
- c. FP (*false positive*) : Yaitu data yang diprediksi positif dan faktanya data itu negatif (Tidak Sesuai).
- d. FN (*false negative*) : Yaitu data yang diprediksi negatif dan faktanya data itu positif (Tidak Sesuai).

Melalui metode *confusion matrix* bisa didapat nilai presisi, akurasi, serta *recall* (Chely Aulia Misrun et al., 2023).

Akurasi diperlukan untuk evaluasi dan mengukur keakuratan dari hasil klasifikasi, semakin besar nilai akurasi maka semakin baik tingkat klasifikasinya (Turmudi Zy et al., 2021). Rumus perhitungan nilai akurasi ditunjukkan pada persamaan 2.5 berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.6)$$

Precision atau dikenal dengan nama *confidence* adalah sebuah model perhitungan untuk mencari hasil dari proporsi jumlah kasus dengan hasil diagnosa positif (Turmudi Zy et al., 2021). Menurut (Febriyani & Februariyanti, 2022), *precision* yaitu perbandingan antara banyaknya data yang diprediksi positif dengan *true positive (TP)*. Rumus perhitungan nilai *precision* ditunjukkan pada persamaan 2.6 berikut:

$$Precision = \frac{TP}{TP+FP} \times 100 \% \quad (2.7)$$

Recall atau *sensitivity* adalah sebuah model perhitungan untuk mencari hasil proporsi jumlah kasus

positif yang diidentifikasi dengan benar (Turmudi Zy et al., 2021). Menurut (Febriyani & Februariyanti, 2022) *recall* yaitu perbandingan antara banyaknya data yang sebenarnya positif dengan *true positive (TP)*. Rumus perhitungan nilai *recall* ditunjukkan pada persamaan 2.7 berikut:

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2.8)$$

F1 Score diperlukan untuk menghitung rata-rata harmonik dari *recall* dan *precision* yang memberikan masing-masing bobot yang sama (Chely Aulia Misrun et al., 2023). Rumus perhitungan nilai *F1 Score* ditunjukkan pada persamaan 2.8 :

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} 100\% \quad (2.9)$$

B. Kajian Pustaka

Tabel 2.2 Penelitian Terdahulu

No	Judul, Penulis, Tahun	Peneliti, Publikasi, Tahun	Hasil Penelitian
1	Analisis Sentimen Pengguna Twitter Terhadap Kinerja Walikota Medan	Fajar Muharram dan Kana Saputra S.,	Hasil dari penelitian ini yaitu <i>Naïve Bayes</i> dengan menggunakan metode SMOTE dengan rasio

No	Judul, Penulis, Tahun	Peneliti, Publikasi, Tahun	Hasil Penelitian
	Menggunakan Metode <i>Naive Bayes Classifier</i> .	Jurnal Sistem Informasi dan Ilmu Komputer, 2023	training dan testing 80:20 memiliki akurasi yang lebih tinggi dibandingkan dengan rasio lainnya dan dibandingkan dengan <i>Naive Bayes</i> tanpa menggunakan SMOTE, berikut akurasi yang dimiliki yaitu sebesar 78 % dan memiliki nilai <i>precision</i> 78%, <i>recall</i> 78%, dan <i>f1-score</i> 77% .
2	Analisis Sentimen Mahasiswa Terhadap Kurikulum Literasi Digital di Universitas Singaperbangsa Karawang Menggunakan <i>Naive Bayes</i>	Luthfiyyah et al., Processor: Jurnal Ilmiah Sistem Informasi, Teknologi Informasi dan Sistem Komputer, 2024	Hasil yang diperoleh yaitu model yang dilatih dengan algoritma <i>Naive Bayes</i> menggunakan pembagian 90% data <i>training</i> dan 10% data <i>testing</i> dengan penerapan ADASYN menjadi model terbaik dengan <i>accuracy</i> 89%, <i>precision</i> 100%, <i>recall</i> 85%, dan <i>f1-score</i> 92%
3	Analisis Sentimen Terhadap Program Kampus Merdeka Pada Twitter Menggunakan Metode <i>Naive Bayes</i> , <i>Union dan Synthetic Minority Over Sampling Technique</i> (SMOTE).	Jefina Tri Kumalasari dan Agustiena Merdekawati, SATIN : Sains dan Teknologi Informasi Journal, 2023	Hasil analisa akumulasi sentimen yaitu untuk mengoptimalkan data yang tidak seimbang digunakan <i>Synthetic Minority Oversampling Technique</i> (SMOTE) menghasilkan keakuratan data sebesar 99.21% sedangkan tanpa SMOTE keakuratan data hanya 77.60%.
4	Perbandingan Metode <i>Resampling</i>	Nurhopiah &	Pada penelitian ini dilakukan eksplorasi terhadap beberapa

No	Judul, Penulis, Tahun	Peneliti, Publikasi, Tahun	Hasil Penelitian
	Pada <i>Imbalanced Dataset</i> Untuk Klasifikasi Komentar Program MBKM,	Magnolia, Jurnal Publikasi Ilmu Komputer Dan Multimedia, 2022	teknik <i>resampling</i> dan menggunakan empat <i>classifier</i> yaitu <i>Random Forest</i> , <i>Logistic Regression</i> , <i>SVM</i> dan <i>MLP</i> . Hasil terbaik didapatkan yaitu <i>oversampling</i> menggunakan SMOTE dan ADASYN menghasilkan peningkatan kinerja yang signifikan Nilai terbaik diperoleh pada model klasifikasi SVM dengan dataset hasil <i>resampling</i> metode SMOTE. Setelah melalui optimasi model dan metode <i>resampling</i> diperoleh nilai <i>F-Measure</i> optimal sebesar 0,9524

Penelitian ini akan melakukan analisis sentimen terkait Kebijakan Program Pema'hadan Tahun 2023 UIN Walisongo Semarang yang belum pernah dilakukan analisis sentimen sebelumnya. Data yang digunakan peneliti merupakan data komentar Instagram. Peneliti membandingkan hasil performa yang diberikan metode *Naïve Bayes Classifier* dengan *oversampling* SMOTE dan metode *Naïve Bayes Classifier* dengan *oversampling* ADASYN. Data diklasifikasikan menjadi dua kelas yakni positif dan negatif. Penelitian ini dijalankan dengan menggunakan bahasa pemrograman *Python*.

BAB III

METODE PENELITIAN

A. Metode Pengumpulan Data

1. Studi Literatur

Pada tahap ini penulis melakukan studi literatur dengan melakukan pengumpulan informasi dan referensi yang berkaitan dengan penelitian ini. Proses pengumpulan informasi dan referensi ini memanfaatkan karangan- karangan ilmiah, skripsi, buku, *website* dan sejenisnya untuk dijadikan sumber rujukan dalam mempelajari materi terkait penelitian seperti materi analisis sentimen, metode *Naïve Bayes Classifier*, dan lain sebagainya. Sumber referensi yang digunakan diperoleh baik secara luring maupun daring.

2. Studi Lapangan

Peneliti melakukan pengamatan pada media sosial Instagram yaitu pada akun Instagram @uinwalisongosemarang. Karena akun tersebut merupakan akun resmi dari UIN Walisongo Semarang yang mana melalui akun tersebut pihak kampus secara resmi menyampaikan pengumuman terkait Kebijakan Program Pema'hadan Tahun 2023 yang menjadi bahasan pada penelitian ini.

B. Perangkat Penelitian

Pada proses implementasi sistem perlu disiapkan lingkungan implementasi untuk menjalankan sistem. Berikut lingkungan implementasi perangkat lunak dan perangkat keras yang peneliti gunakan :

1. Perangkat keras :

Tabel 3.1 Kebutuhan Perangkat Keras

No	Perangkat keras	Spesifikasi
1.	<i>Device</i>	Laptop Acer Aspire A314-31-C08A
2.	<i>Processor</i>	Intel Celeron N3350
3.	Memori (RAM)	4 GB
4.	Monitor	14 inch

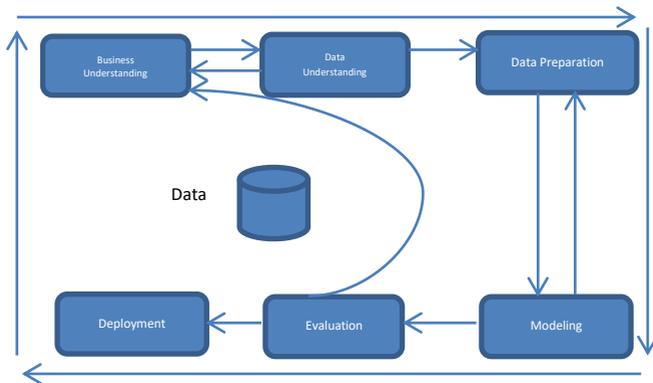
2. Perangkat lunak :

Tabel 3.2 Kebutuhan Perangkat Lunak

No	Perangkat lunak	Spesifikasi
1.	Sistem Operasi	Windows 10 Home
2.	Bahasa Pemrograman	Python
3.	<i>Microsoft Office</i>	Ms. Word, Ms Excell
4.	<i>Browser</i>	<i>Chrome</i>
5.	<i>Software</i>	<i>Jupyter Notebook</i>

C. Alur Pengerjaan Penelitian

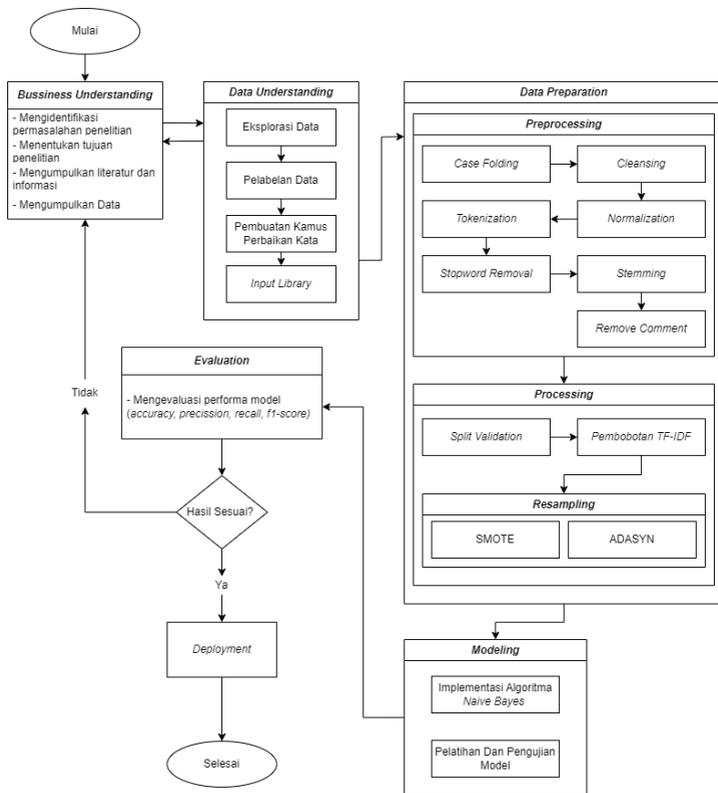
Metode dalam melakukan analisis sentimen ini adalah dengan menggunakan metode *Cross Industry Standard Process for Data Mining* (CRISP-DM). Penelitian ini menggunakan CRISP-DM dalam konteks analisis sentimen untuk menghasilkan informasi yang valid dan kredibel sebagaimana tahapan-tahapan dalam metode CRISP-DM (Y A Singgalen, 2023). Tahapan pada metode CRISP-DM diantaranya yaitu tahapan *business understanding*; tahapan *data understanding*; tahap *data preparation*; tahap *modeling*; tahap *evaluation*; dan tahap *deployment* (Afrianto Singgalen et al., 2023). Berikut adalah alur gambaran dari metode CRISP-DM.



Gambar 3. 1 Alur Metode CRISP-DM

Gambar diatas merupakan merupakan bagan alur kerja dari keseluruhan tahapan metode CRISP-DM untuk menganalisis data dari komentar pada salah satu unggahan pada Instagram

UIN Walisongo Semarang yaitu @uinwalisongosemarang dengan tujuan untuk dapat mengetahui permasalahan bisnis yang terjadi pada Kebijakan Program Pema'hadan Tahun 2023 UIN Walisongo Semarang sehingga dapat dijadikan sebagai bahan evaluasi bagi kampus dalam mengoptimalkan penerapan kebijakan yang dibuat. Tahapan dengan menggunakan CRISP-DM mengacu pada diagram dibawah ini.



Gambar 3. 2 Diagram Alur Penelitian

Secara rinci tahapan pada metode CRISP-DM adalah sebagai berikut :

1. *Bussines Understanding*

Tahap yang pertama dilakukan adalah tahapan *business understanding*. Tahap ini dimulai dengan memahami masalah bisnis yang ingin diselesaikan atau tujuan yang ingin dicapai dengan proyek *data mining*. Pada penelitian ini Kebijakan Program Pema'hadan Tahun 2023 UIN Walisongo Semarang menjadi fokus utama dalam penelitian ini. Pengumuman kebijakan tersebut disampaikan malalui akun Instagram resmi UIN Walisongo Semarang yaitu @uinwalisongosemarang sehingga Instagram menjadi *platform* yang digunakan dalam pengambilan data (*text mining*).

Pengambilan data dilakukan dengan *export* melalui ekstensi *chrome* yaitu IGCommentExporter. Tujuan pengumpulan respon pada komentar tersebut yaitu agar dapat diklasifikasikan menjadi sentimen positif maupun negatif serta dianalisis aspek-aspek yang dibahas pada komentar tersebut sehingga dapat dijadikan sebagai bahan evaluasi untuk meningkatkan pelayanan dan sebagai bahan pertimbangan dalam pengambilan kebijakan kedepannya.

2. *Data Understanding*

Pada tahap ini, data yang tersedia dieksplorasi dan dipahami secara mendalam. Hal ini mencakup pemeriksaan dan pemahaman data yang tersedia, struktur data, kualitas data, serta kemungkinan masalah (Fitrianti et al., 2023).

Data yang digunakan di ambil dari data komentar Instagram sebanyak 565 data. Format data yang digunakan yaitu format csv. Berikut contoh hasil pengambilan data Instagram sebagai berikut.

Tabel 3.3 Contoh Hasil Pengambilan Data Instagram

Username	Komentar
user001	mandek mondok pengen kuliah, eh malah dikon mondok meneh anjirrr
user002	@imchoice19__ betuuulll
user003	Akwoksowk welcome to ta'dziran tiap minggu
user004	Itu segitu ada wajib bayar makan lagi apa ga?
user005	@heni_fatmaa belum.. ada bayar katering dan laundry
user006	@marco_marconah serius?
user007	@marco_marconah sekitar berapa?
user008	@wahyunugrohosp katering nya 450 k/ bulan terus laundry nya per kg 7 k
user009	@wahyunugrohosp Ya gak tau wkwkwk heheheh bisa2 intinya sabar aku juga pemaksaan kok
user010	@nittcollage gmn komentarnya soal makanannya?

Dilanjutkan dengan proses pelabelan data untuk menentukan opini/tanggapan yang ada. Pelabelan yang dilakukan yakni melabeli data dengan 2 kategori, positif dan negatif. Contoh hasil dari proses pelabelan adalah sebagai berikut:

Tabel 3.4 Contoh Hasil Pelabelan

Opini	Label
Pemaksaan!!!!!! Sistemnya, bnyk camaba mengundurkan diri grgr dsuruh ma'had	Negatif
menurut gw fasilitas nya udh bagus kok, banyak yg sudah upgrade	Positif

3. *Data Preparation*

Pada tahap ini data disiapkan untuk digunakan dalam proses *data mining*. Hal ini meliputi pemilihan atribut yang relevan, transformasi data, dan eliminasi data yang tidak relevan atau tidak valid.

Tujuan dari tahap ini agar data yang masih mentah dapat dibersihkan sehingga bisa dilakukan pengklasifikasian. Tahapan *data preparation* ini dibagi menjadi 2 tahap yaitu *preprocessing* dan *processing*.

a. *Preprocessing*

1). *Case folding*

Pada tahap ini, suatu kata, kalimat atau teks yang mengandung huruf kapital disamaratakan menjadi huruf kecil atau *lowercase*. Tujuannya adalah untuk menyamakan/menyeragamkan bentuk kata. Contoh proses *data preparation* adalah seperti dibawah ini:

Tabel 3.5 Contoh Hasil *Case Folding*

<i>Input Process</i>	<i>Output Process</i>
Saya salah satu dari cam aba, dan dengan keputusan tsb saya langsung men gundurkan diri hehe..	saya salah satu dari cam ba, dan dengan keputusa n tsb saya langsung men gundurkan diri hehe..

2). *Cleansing*

Data komentar biasanya masih terdapat nama username ketika melakukan *reply* komentar dari akun lain dan tag kepada akun lain seperti (@Username). Selain itu masih terdapat URL (*Uniform Resource Locator*), bilangan angka, symbol, dan tanda baca membuat data menjadi tidak digunakan. Karena komponen-komponen tersebut tidak memiliki peran apapun dalam

melakukan sentimen maka secara sistem akan dihapus. Tujuannya yaitu untuk mengurangi *noise*. Berikut contoh dari proses *cleansing*:

Tabel 3.6 Contoh Hasil *Cleansing*

<i>Input Process</i>	<i>Output Process</i>
saya salah satu dari camaba, dan dengan keputusan tsb saya langsung men gundurkan diri hehe..	saya salah satu dari camaba dan dengan keputusan tsb saya langsung men gundurkan diri hehe

3). *Normalization*

Tentunya pada pengguna dari data komentar Instagram tidak heran jika para penggunanya menulis kata dengan singkat-singkat, kata yang tidak baku atau yang tidak formal membuat suatu opini dengan menggabungkan dua bahasa yaitu Indonesia-Inggris sekaligus dalam satu kalimat atau bahkan menggunakan bahasa daerah seperti Bahasa Jawa. Nah, pada tahap *normalization* dilakukan untuk pengubahan kata yang tidak sesuai dengan KBBI. Contoh penerapan proses *normalization* adalah sebagai berikut :

Tabel 3.7 Contoh Hasil *Normalization*

<i>Input Process</i>	<i>Output Process</i>
saya salah satu dari cama ba dan dengan keputusa n tsb saya langsung men gundurkan diri hehe	saya salah satu dari calon mahasiswa baru dan den gan keputusan tersebut s aya langsung mengundur kan diri hehe

4). *Tokenization*

Tokenization adalah tahap memecah kalimat menjadi bagian-bagian kata yang dinamakan token. Pada proses ini akan menghilangkan *whitespace*. Berikut penerapan proses *Tokenization*:

Tabel 3.8 Contoh Hasil *Tokenization*

<i>Input Process</i>	<i>Output Process</i>
saya salah satu dari calon mahasiswa baru dan den gan keputusan tersebut sa ya langsung mengundurk an diri hehe	['saya', 'salah', 'satu', 'dari', ' ', 'calon', 'mahasiswa', 'ba ru', 'dan', 'dengan', 'keput usan', 'tersebut', 'saya', 'la ngsung', 'mengundurkan', 'diri', 'hehe']

5). *Stopword Removal*

Pada tahap *stopword removal* dilakukan untuk menghilangkan atau menghapus daftar kata umum yang tidak memiliki arti penting atau informasi yang dibutuhkan. Tujuannya adalah

untuk mengurangi jumlah kata yang disimpan oleh sistem. Berikut contoh dari penerapan proses *stopword removal*:

Tabel 3.9 Contoh Hasil *Stopword Removal*

<i>Input Process</i>	<i>Output Process</i>
['saya', 'salah', 'satu', 'dari', 'calon', 'mahasiswa', 'baru', 'dan', 'dengan', 'keputusan', 'tersebut', 'saya', 'langsung', 'mengundurkan', 'diri', 'hehe']	['salah', 'calon', 'mahasiswa', 'keputusan', 'langsung', 'mengundurkan']

6). *Stemming*

Pada tahap *stemming* untuk mengganti suatu kata yang memiliki imbuhan menjadi bentuk kata dasar, dan juga mengubah kata perulangan kata menjadi satu kata dasar. Berikut kaidah yang telah dibuat dalam proses *stemming* pada bahasa Indonesia (Luqyana, 2018), antara lain:

- a) Prefiks, imbuhan yang terletak pada awal kata. Prefiks terdiri dari “se-“, ”ke”, ”me-“ dll. Contoh: Me-rasa
- b) Suffiks, imbuhan yang terletak pada akhir kata. Contoh dari suffiks adalah “-lah”, “-kah”, “-pun” dll. Contoh: turun-kan.

- c) Konfiks, imbuhan ini merupakan imbuhan gabungan dari prefiks dan suffiks. Imbuhan terdapat pada awal dan akhir kata. Contoh: perasa-an.
- d) Infiks, imbuhan yang terletak pada tengah kata. Contoh: k-em-ilau dari kata kilau.
- e) Perulangan kata, contoh: anak-anak.

Contoh penerapan *stemming* pada data adalah sebagai berikut:

Tabel 3.10 Contoh Hasil *Stemming*

<i>Input Process</i>	<i>Output Process</i>
['salah', 'calon', 'mahasiswa', 'keputusan', 'langsung', 'mengundurkan']	salah calon mahasiswa putus langsung undur

7). *Remove Comment*

Langkah selanjutnya yaitu proses penghapusan data komentar yang kosong atau tidak memiliki isi. Langkah ini diterapkan dengan menggunakan beberapa perintah kode yang berfungsi untuk menghapus komentar yang kosong.

b. *Processing*

Tidak berhenti disitu, masih terdapat langkah selanjutnya dari tahap ini yang disebut *processing*. Tahap *processing* dilakukan beberapa tahap diantaranya :

1). *Split Validation*

Data yang telah melalui *preprocessing* selanjutnya dilakukan langkah *split validation* yang mana data dibagi antara data latih dengan data uji dengan rasio 90:10 dimana 90% dari keseluruhan jumlah data dijadikan data latih dan 10% dari keseluruhan data dijadikan data uji yang diambil secara acak.

2). Pembobotan TF-IDF

Setelah dibagi, kemudian dilakukan pembobotan nilai kata dengan menggunakan TF-IDF untuk mempermudah jalannya proses pengklasifikasian *Naïve Bayes* .

3). *Resampling*

Setelah dilakukan pembobotan, selanjutnya data melalui proses *resampling*. Tujuan *resampling* adalah memodifikasi dataset sehingga lebih proporsional atau lebih representatif.

Pengolahan algoritma yang tidak mempertimbangkan ketidakseimbangan data cenderung menitikberatkan kelas mayor dan bukan kelas minor, oleh sebab itu diperlukan *oversampling* untuk memperbanyak pengamatan secara acak dengan menambah jumlah data kelas minor (data buatan) agar setara dengan kelas mayor (Yerik Afrianto Singgalen, 2023).

4. *Modeling*

Tahap ini mencakup pemilihan model *data mining* yang sesuai untuk menyelesaikan masalah bisnis yang telah diidentifikasi. Setelah dilakukan *data preparation* barulah masuk pada tahapan yang menjadi intinya yaitu pengklasifikasian dengan metode *Naïve Bayes*. Metode *Naïve Bayes* merupakan metode yang digunakan untuk mengklasifikasikan data komentar untuk memperoleh sentimen analisis.

Pengklasifikasian dibagi menjadi dua kelas yaitu positif dan negatif. Uji model sendiri dilakukan untuk mengetahui kinerja model. Setelah uji model dilakukan maka akan tampil seberapa besar performa metode yang dilakukan.

5. *Evaluation*

Tahap ini mengevaluasi semua tahapan yang telah dilakukan berdasarkan tujuan semula. Setelah uji model dilakukan, lalu pada tahap evaluasi model untuk menentukan tingkat keakurasian, *precisson*, *recall*, dan *f1 score* melalui metode *confusion matrix* pada tabel dengan tujuan mengetahui performa dari model tersebut.

Evaluasi model dilakukan dengan cara melihat tingkat performa metode melalui *confusion matrix*. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya. Lebih tepatnya, data *test* diujikan terhadap data *training*, maka akan menghasilkan daftar kelas-kelas dari data *test*, sebut saja prediksi kelas. Kemudian prediksi kelas dibandingkan dengan kelas yang sebenarnya dari data *test* yang disembunyikan sebelumnya. Sehingga dapat dilihat pada performa model *Naïve Bayes* yang berupa tingkat akurasi, *pressicion*, *recall* dan *f1 score* .

6. *Deployment*

Pada tahap ini, model *data mining* yang telah dikembangkan diimplementasikan ke dalam

lingkungan produksi untuk digunakan dalam proses bisnis.

Tahap akhir yaitu memvisualisasikan hasil dari analisis sentimen menggunakan diagram lingkaran dan *wordcloud*. *Wordcloud* akan digunakan untuk memvisualisasikan hasil analisis klasifikasi. Tujuan dari visualisasi tersebut adalah untuk mengekstrak informasi berupa topik yang sering dibahas yaitu Kebijakan Program Pema'hadan Tahun 2023 UIN Walisongo Semarang sehingga dapat diambil informasi yang dianggap penting dari sekian banyak teks tanggapan yang ada.

BAB IV

HASIL DAN PEMBAHASAN

A. *Bussines Understanding*

Tahap yang pertama dilakukan adalah tahapan *business understanding*. Tahap ini dimulai dengan memahami masalah bisnis yang ingin diselesaikan atau tujuan yang ingin dicapai dengan proyek *data mining*. Pada penelitian ini Kebijakan Program Pema'hadan Tahun 2023 UIN Walisongo Semarang menjadi fokus utama dalam penelitian ini. Pengumuman kebijakan tersebut disampaikan malalui akun Instagram resmi UIN Walisongo Semarang yaitu @uinwalisongosemarang sehingga Instagram menjadi *platform* yang digunakan dalam pengambilan data (*text mining*). Permasalahan yang ditemukan pada data tersebut yakni belum diketahui bagaimana arah respon masyarakat terhadap kebijakan tersebut apakah positif atau negatif. Tujuan dari penelitian ini yakni untuk mengetahui arah respon masyarakat terhadap kebijakan tersebut dengan menerapkan algoritma *Naïve Bayes* dan mengetahui bagaimana performa yang diberikan metode tersebut.

Pengambilan data dilakukan dengan cara mengekspor data melalui ekstensi *chrome* yaitu IGCommentExporter. IGCommentExporter merupakan alat yang dibuat untuk memudahkan pengguna internet dalam mengekspor komentar Instagram ke dalam bentuk CSV dengan cepat hanya dengan satu

klik. Cara menggunakan IGCommentExporter ini sangat mudah yakni hanya dengan memasukkan URL postingan Instagram kemudian klik tombol ekspor. Data yang dapat diekspor diantaranya *comment id*, *comment text*, *username*, *username id*, *profile url*, *profile pic url*, *date*. Berikut adalah tampilan dari IGCommentExporter pada gambar 4.1 dibawah ini.



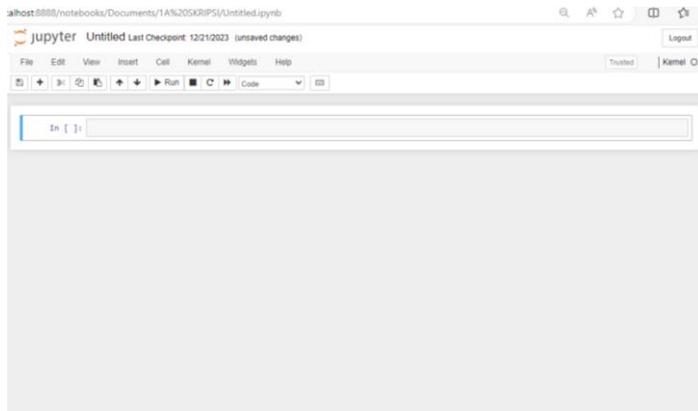
Gambar 4. 1 Tampilan IGCommentExporter

Setelah memasukkan tautan dan kemudian diekspor data akan otomatis terunduh. Gambar 4.2 menunjukkan contoh data yang terunduh dari IGCommentExporter.

User Id	Username	Comment Id	Comment Text	Profile URL	Profile Pic URL	Date
2250471333	baharuddin_29	180857149543385:	@fauzansidik05 benar sekali mas, setuju sama pe	https://www.https://instagram/5/1/2023,		1:21:55 AM
4240985163	elisaaeu	179741558722061:	@wrlldzar4 sama kak aku juga segitu, bingung bgt	https://www.https://instagram/5/1/2023,		5:28:43 PM
46762126832	nabilath_	180106292266189:	@wrlldzar4 sama kakð™c	https://www.https://instagram/5/2/2023,		12:34:37 AM
7163238623	bunayffh	180155626485866:	"@notafreyork itu kak, mungkin niat dari kampu	https://www.https://instagram/5/2/2023,		7:15:27 PM
3636243231	shella_aprilyana	182158050042329:	Jadi kangen Ma'had nggak siih? ð™", Apalagi pas k	https://www.https://instagram/5/2/2023,		7:56:43 PM
3635948125	shellyapriyani02	179418743606416:	"@shella_aprilyana ma'had sangat membantu p	https://www.https://instagram/5/2/2023,		8:04:44 PM
3635948125	shellyapriyani02	180009691487732:	@shellyapriyani02_ð™", @azizahrmufidah gimar	https://www.https://instagram/5/2/2023,		8:06:28 PM
3734525348	azizahrmufidah	178650022348821:	@shellyapriyani02_ð™jelaass donggl jump	https://www.https://instagram/5/2/2023,		8:08:46 PM

Gambar 4. 2 Contoh Data Hasil Ekspor

Pengolahan data dilakukan dengan menggunakan aplikasi Jupyter Notebook. Jupyter Notebook adalah aplikasi web gratis yang digunakan oleh *data scientist* untuk membuat dan membagikan dokumen yang berisi kode, proses perhitungan, analisis, visualisasi, dan pengolahan data lainnya. Jupyter Notebook memiliki dukungan untuk lebih dari 40 bahasa pemrograman, termasuk Python. Python menjadi bahasa pemrograman yang digunakan untuk melakukan pengolahan data pada penelitian ini. Gambar 4.3 berikut ini merupakan tampilan dari aplikasi Jupyter Notebook.



Gambar 4. 3 Tampilan Aplikasi Jupyter Notebook

B. Data Understanding

Pada tahap ini, data yang tersedia dieksplorasi dan dipahami secara mendalam. Hal ini mencakup pemeriksaan dan pemahaman data yang tersedia, struktur data, kualitas data, serta kemungkinan masalah (Fitrianti et al., 2023). Data hasil ekspor

yang diperoleh berjumlah 565 baris data dan memiliki beberapa atribut diantaranya *user id*, *username*, *comment id*, *comment text*, *profile URL*, *profile pic URL*, dan *date*. Tidak semua atribut tersebut digunakan sehingga perlu dihapus dan menyisakan atribut yang digunakan yaitu *comment text*.

Langkah selanjutnya yaitu proses pelabelan, data komentar tersebut diberikan label positif untuk komentar yang setuju atau pro dengan Kebijakan Program Pemahadan UIN Walisongo Tahun 2023 dan label negatif untuk komentar yang tidak setuju atau kontra dengan kebijakan program pemahadan UIN Walisongo tahun 2023. Pelabelan dilakukan oleh seorang yang ahli dalam bidang Bahasa Indonesia. Setelah diberikan label, data menjadi memiliki 2 atribut yaitu *comment text*, yang kemudian diubah namanya menjadi komentar, dan label. Berikut adalah contoh data yang sudah diberi label yang ditunjukkan oleh gambar 4.4.

Komentar	Label
mandek mondok pengen kuliah, eh malah dikon mondok meneh anjirrr	negatif
@imchoice19__betuuulll	negatif
Akwoksowk welcome to ta'dziran tiap minggu	negatif
Itu segitu ada wajib bayar makan lagi apa ga?	positif
@heni_fatmaa belum.. ada bayar catering dan laundry	positif
@marco_marconah serius?	negatif
@marco_marconah sekitar berapa?	positif
@wahyunugrohosp catering nya 450 k/ bulan terus laundry nya per kg 7 k	positif
@wahyunugrohosp Ya gak tau wkwwk heheheh bisa2 intinya sabar aku juga pema	negatif
@nittcollage gmn komentarnya soal makanannya?	positif
@marco_marconah ya gitulah wkwwk???	negatif
@marco_marconah makanan nya enk banget pokoknya mana nasi nya banyak lagi?	positif
@dwiilitann mantap... katanya kalau lauk ayam, ayamnya separoh ya? Tp yg belur	negatif
@heni_fatmaa btw gk di sediakan kompor juga kah di Ma'had?	positif
@ainunsiregar_09 mahad engga ada dapur nya, tp setidaknya ada dispenser sm air	positif
@dwiilitann woiilah intan intan jujur sekali kauuu ??????	negatif
@dwiilitann weh nyasanya jauh bener	negatif
@heni_fatmaa tetep bayar makan dong sama londry	positif
@nittcollage biava catering 450 sehari makan brp x kak	positif

Gambar 4. 4 Contoh Data Berlabel

Pada penelitian ini pemrosesan data dilakukan dengan dasar Bahasa Indonesia, namun komentar terkait Kebijakan Program Pema'hadan ini masih terdapat penulisan yang menggunakan bahasa asing dan penulisan dengan singkatan, sehingga bahasa pada komentar tersebut perlu diseragamkan menjadi Bahasa Indonesia yang baku. Penyesuain penulisan ini dilakukan dengan menggunakan kamus yang memuat tulisan yang tidak baku disertai dengan perbaikannya. Pembuatan kamus tersebut dikakukan secara manual dengan data kamus berformat CSV. Peneliti membuat 2 kamus perbaikan kata yang pertama bernama “*Kamus bahasa tidak baku.csv*” dan yang kedua bernama “*kata tidak baku 2.csv*”. Contoh kamus yang digunakan untuk perbaikan kata disajikan pada gambar 4.5 berikut.

Tidak Baku	Baku	
aamiin	amin	
about	tentang	
account	akun	
actually	sebenarnya	
adekk	adik	
adeku	adikku	
adick	adik	
adiks	adik	
adix	adik	
afaan	apaan	
ajaaaa	aja	
akwoksowk	tertawa	

Gambar 4. 5 Contoh Kamus Perbaikan

Data-data yang sudah dimiliki selanjutnya diproses dengan aplikasi Jupyter Notebook menggunakan bahasa pemrograman Python. Sebelum melakukan pemrosesan perlu disiapkan *library-library* pendukung seperti pada gambar 4.6 berikut.

```
In [1]: import string
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import nltk
```

Gambar 4. 6 Kode Pemanggilan *Library*

Gambar 4.6 menunjukkan pendeklarasian *library string*, *library pandas*, *library matplotlib*, *library seaborn*, dan *library NLTK*. *Library string* dalam Python digunakan untuk melakukan berbagai operasi terkait dengan manipulasi *string* seperti pemformatan, pencarian, penggantian, pengubahan ukuran huruf dan lain-lain. *Library pandas* adalah *library* pada Python yang berguna untuk pengolahan data. *Library matplotlib* adalah *library* visualisasi data pada Python yang dapat digunakan untuk membuat berbagai jenis grafik, plot, dan visualisasi data. *Library seaborn* merupakan *library* Python yang dibangun diatas *library matplotlib* dan dirancang khusus untuk visualisasi data statistik. *Library NLTK (Natural Language Toolkit)* akan digunakan dalam tahap *preprocessing data* seperti *stopwords removal*, dan *tokenization*.

C. *Data Preparation*

Setelah mengimpor *library* yang dibutuhkan selanjutnya data menuju tahap *data preparation*. Tahapan ini data disiapkan untuk digunakan dalam proses *data mining*. Data tersebut masih terdapat banyak *noise* yang perlu dihilangkan untuk proses klasifikasi agar menjadi lebih efisien dan akurat (Dedy et al., 2023). Tujuan dari tahap ini agar data yang masih mentah dapat dibersihkan sehingga bisa dilakukan pengklasifikasian. Terdapat dua langkah dalam *data preparation* ini yaitu *preprocessing* dan *processing*.

1. *Preprocessing*

Adapun urutan dari tahapan *preprocessing* yang dilakukan adalah sebagai berikut:

a). *Case Folding*

Noise yang terdapat pada data salah satunya yaitu penulisan masih tertulis dengan menggunakan huruf kapital dan huruf kecil, maka dari itu data perlu diseragamkan hurufnya menjadi huruf kecil atau *lowercase* karena Python akan menganggap berbeda antara huruf kapital dan huruf kecil meskipun secara bahasa alami memiliki pengucapan dan makna yang

sama. Kode untuk proses *case folding* ditunjukkan oleh gambar 4.7.

Case Folding

```
def casefolding(komentar):
    komentar = komentar.lower()
    return komentar
data['Lower'] = data['Komentar'].apply(casefolding)
data.head()
```

Gambar 4.7 Kode Penerapan Proses *Case Folding*

Hasil penerapan *case folding* ditunjukkan pada gambar berikut ini.

	Komentar	Label	Lower
0	mandek mondok pengen kuliah, eh malah dikon mo...	negatif	mandek mondok pengen kuliah, eh malah dikon mo...
1	@imchoice19__betuuulll	negatif	@imchoice19__betuuulll
2	Akwoksowk welcome to ta'dziran tiap minggu	negatif	akwoksowk welcome to ta'dziran tiap minggu
3	itu segitu ada wajib bayar makan lagi apa ga?	positif	itu segitu ada wajib bayar makan lagi apa ga?
4	@heni_fatmaa belum.. ada bayar katering dan la...	positif	@heni_fatmaa belum.. ada bayar katering dan la...

Gambar 4. 8 Hasil Penerapan *Case Folding*

b). *Cleansing*

Pada tahap *cleansing* dilakukan beberapa penghapusan komponen diantaranya *username* (@nama), URL (*Uniform Resource Locator*), bilangan angka, simbol, tanda baca, emoji, dan lain-lain. Karena komponen-komponen tersebut tidak memiliki peran apapun dalam melakukan sentimen maka secara sistem akan dihapus.

Tahap *cleansing* ini memerlukan *library* pendukung yaitu *library re* dan *library emoji*. *Library*

re berfungsi untuk melakukan tahapan *Regular Expression (regex)* atau deretan karakter yang digunakan untuk pencarian teks dengan menggunakan pola (*pattern*). *Library emoji* yang bertujuan untuk mengubah emoji menjadi suatu teks. Gambar 4.9 menunjukkan kode untuk mengimpor *library re* dan *library emoji*.

```
import re, emoji
```

Gambar 4. 9 Kode Pemanggilan *Library Re* Dan *Emoji*

Adapun kode program untuk pengimplementasian tahap *cleansing* dapat dilihat pada gambar 4.10.

Cleansing

```
def cleansing(komentar):  
    komentar = komentar.strip(" ")  
    komentar = re.sub(r'[?!\.!\!]+(?=\.\!])', '', komentar)  
    komentar = re.sub('\B@w+', '', komentar)  
    komentar = re.sub(r'[^a-zA-Z]', ' ', komentar)  
    komentar = emoji.demojize(str(komentar))  
    komentar = komentar.strip(" ")  
    return komentar  
data['Clean'] = data['Lower'].apply(cleansing)  
data.head(10)
```

Gambar 4. 10 Kode Penerapan Proses *Cleansing*

Hasil penerapan *cleansing* ditunjukkan pada gambar 4.11 berikut ini.

Lower	Clean
mandek mondok pengen kuliah, eh malah dikon mo...	mandek mondok pengen kuliah eh malah dikon mo...
@imchoice19__betuuull	betuuull
akwoksovkw welcome to ta'dziran tiap minggu	akwoksovkw welcome to ta dziran tiap minggu
itu segitu ada wajib bayar makan lagi apa ga?	itu segitu ada wajib bayar makan lagi apa ga
@heni_fatmaa belum... ada bayar katering dan la...	belum ada bayar katering dan laundry
@marco_marconah serius?	serius
@marco_marconah sekitar berapa?	sekitar berapa
@wahyunugrohosp katering nya 450 k/ bulan teru...	katering nya k bulan terus laundry nya pe...
@wahyunugrohosp ya gak tau wkwkwkw heheheh bisa...	ya gak tau wkwkwkw heheheh bisa intinya sabar ...
@niticollage gmn komentarnya soal makanannya?	gmn komentarnya soal makanannya

Gambar 4. 11 Hasil Penerapan *Cleansing*

c). *Normalization*

Tahap *normalization* dilakukan untuk pengubahan kata yang tidak sesuai (singkatan, kata tidak baku, dll.) sesuai dengan KBBI.

Kamus perbaikan kata yang sudah dibuat pada tahap *data understanding*, kemudian digunakan pada tahap *normalization* ini. Kamus perbaikan kata dipanggil terlebih dahulu seperti yang ditampilkan pada gambar 4.12 dan 4.13 berikut.

```
kamus_slang = pd.read_csv('Kamus bahasa tidak baku.csv', sep=',')
slang = pd.Series(kamus_slang['Baku'].values, index = kamus_slang['Tidak Baku']).to_dict()
kamus_slang.head()
```

Tidak Baku	Baku
0	aamin amin
1	about tenlang
2	account akun
3	actually sebenarnya
4	adekk adik

Gambar 4. 12 Kode Pemanggilan Kamus Perbaikan Kata 1

```
kamus_slang2 = pd.read_csv('kata tidak baku 2.csv')
slang2 = pd.Series(kamus_slang2['makna'].values, index = kamus_slang2['slang']).to_dict()
kamus_slang2.head()
```

	slang	makna
0	aaa	apa
1	abis	habis
2	ad	ada
3	afah	apa
4	aga	agak

Gambar 4. 13 Kode Pemanggilan Kamus Perbaikan Kata 2

Selanjutnya proses penerapan kamus perbaikan kata untuk mengganti kata yang tidak sesuai dilakukan dengan menjalankan kode seperti pada gambar 4.14 dan 4.15 berikut.

```
def Slangwords(komentar):
    for word in komentar.split():
        if word in slang.keys():
            komentar = komentar.replace(word, slang[word])
            komentar = re.sub('@[\w]+', '', komentar)
    return komentar
data['Slang1'] = data['Clean'].apply(Slangwords)
data.to_csv('data_normal.csv')
data.head()
```

Gambar 4. 14 Kode Penerapan *Normalization 1*

```
def Slangwords(komentar):
    for word in komentar.split():
        if word in slang2.keys():
            komentar = komentar.replace(word, slang2[word])
            komentar = re.sub('@[\w]+', '', komentar)
    return komentar
data['Slang2'] = data['Slang1'].apply(Slangwords)
data.to_csv('data_normal2.csv')
data.head()
```

Gambar 4. 15 Kode Penerapan *Normalization 2*

Hasil dari penerapan normalization dapat dilihat pada gambar 4.16 berikut.

	Komentar	Label	Lower	Clean	Slang1	Slang2
0	mandek mondok pengen kuliah, eh malah dikon mo...	negatif	mandek mondok pengen kuliah, eh malah dikon mo...	mandek mondok pengen kuliah eh malah dikon mo...	berhenti pondok ingin kuliah eh malah disuruh...	berhenti pondok ingin kuliah eh malah disuruh...
1	@imchoice19__betulull	negatif	@imchoice19__betulull	betulull	betul	betul
2	Akroksokw welcome to ta'dzran bap mnggu	negatif	akroksokw welcome to ta'dzran bap mnggu	akroksokw welcome to ta'dzran bap mnggu	tertawa selamat datang di ta' huluman setiap m...	tertawa selamat datang di ta' huluman setiap m...
3	itu segitu ada wajib bayar makan lagi apa ga?	positif	itu segitu ada wajib bayar makan lagi apa ga?	itu segitu ada wajib bayar makan lagi apa ga	itu segitu ada wajib bayar makan lagi apa ga	itu segitu ada wajib bayar makan lagi apa ga
4	@heni_fatmaa belum ada bayar katering dan la...	positif	@heni_fatmaa belum ada bayar katering dan la...	belum ada bayar katering dan laundry	belum ada bayar katering dan penata	belum ada bayar katering dan penata

Gambar 4. 16 Hasil Penerapan *Normalization*

d). *Tokrnization*

Data komentar yang sudah dinormalisasi selanjutnya dipecah menjadi token-token atau bagian-bagian kata yang disebut dengan tahapan *tokenization*. *Tokenization* penting dilakukan agar memudahkan komputer dalam memahami teks. *Tokenization* dilakukan dengan melakukan *split*, berikut kode pengimplementasian *tokenization* pada gambar 4.17.

```
def token(text):
    tokenisasi = text.split()
    return tokenisasi
data['Tokenize'] = data['Slang2'].apply(token)
data.head()
```

Gambar 4.17 Kode Penerapan *Tokenization*

Setelah diterapkan *tokenization* kalimat-kalimat pada data tersebut akan menjadi pecahan kata seperti pada gambar 4.18 dibawah ini.

Slang1	Slang2	Tokenize
berhenti pondok ingin kuliah eh malah disuruh...	berhenti pondok ingin kuliah eh malah disuruh...	[berhenti, pondok, ingin, kuliah, eh, malah, d...]
betul	betul	[betul]
tertawa selamat datang di ta hukuman setiap mi...	tertawa selamat datang di ta hukuman setiap mi...	[tertawa, selamat, datang, di, ta, hukuman, se...]
itu segitu ada wajib bayar makan lagi apa ga	itu segitu ada wajib bayar makan lagi apa ga	[itu, segitu, ada, wajib, bayar, makan, lagi, ...]
belum ada bayar katering dan penatu	belum ada bayar katering dan penatu	[belum, ada, bayar, katering, dan, penatu]

Gambar 4.18 Hasil Implementasi *Tokenization*

e). *Stopword Removal*

Stopword removal dilakukan untuk menghilangkan atau menghapus daftar kata umum yang tidak memiliki arti penting atau informasi yang dibutuhkan. Tujuannya adalah untuk mengurangi jumlah kata yang disimpan oleh sistem.

Library NLTK (Natural Language Toolkit) diperlukan dalam proses ini, karena *library* ini menyediakan modul *stopwords* yang memuat daftar kata-kata penghenti (*stop words*) dalam beberapa bahasa termasuk Bahasa Indonesia. Modul *stopwords* perlu dipanggil terlebih dahulu seperti pada gambar 4.19 berikut ini.

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
```

Gambar 4. 19 Pemanggilan Modul *Stopwords*

Selain menggunakan daftar kata dari modul *stopwords*, penulis menambahkan daftar *stopword* tabahan yang bernama “*stopwords.txt*”, berisi kata tidak sesuai yang masih terdapat pada data. Kode penerapan *stopword removal* seperti pada gambar 4.20.

Stopword Removal

```
stop = stopwords.words('indonesian')
txt_stopword = pd.read_csv("stopwords.txt", names= ["stopwords"], header = None)
stop.extend(txt_stopword["stopwords"][0].split(' '))

def stopwords (hasil_token):
    return [word for word in hasil_token if word not in stop]
data['Stopword'] = data['Tokenize'].apply(lambda x: stopwords(x))
data.to_csv('data_stopword.csv')
data.head()
```

Gambar 4. 20 Kode Penerapan *Stopword Removal*

Setelah di terapkan *stopword removal* hasilnya ditunjukkan pada gambar 4.21 ini.

Slang2	Tokenize	Stopword
berhenti pondok ingin kuliah eh malah disuruh...	[berhenti, pondok, ingin, kuliah, eh, malah, d...]	[berhenti, pondok, kuliah, disuruh, pondok]
betul	[betul]	[]
tertawa selamat datang di ta hukuman setiap mi...	[tertawa, selamat, datang, di, ta, hukuman, se...]	[tertawa, selamat, hukuman, minggu]
itu segitu ada wajib bayar makan lagi apa ga	[itu, segitu, ada, wajib, bayar, makan, lagi, ...]	[segitu, wajib, bayar, makan]
belum ada bayar katering dan penatu	[belum, ada, bayar, katering, dan, penatu]	[bayar, katering, penatu]

Gambar 4. 21 Hasil Penerapan *Stopword Removal*

f). *Stemming*

Stemming dilakukan untuk mengganti suatu kata yang memiliki imbuhan dan kata perulangan menjadi satu kata dasar. *Stemming* dilakukan dengan menambahkan *library* yang bernama *library* sastrawi. *Library* sastrawi merupakan sebuah perpustakaan kode yang dapat digunakan dalam pemrosesan bahasa alami dalam Bahasa Indonesia. Pada tahap ini *library* sastrawi digunakan untuk penyederhanaan kata dengan modul *stemmer factory* yang termuat didalamnya. Maka dari itu modul *stemmerfactory* dari *library* sastrawi perlu ditambahkan dengan menggunakan kode seperti dibawah ini.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

Gambar 4. 22 Kode Pemanggilan *Modul Stemmer Factory*

Selanjutnya modul *stemmer factory* digunakan dalam proses *stemming* seperti yang gambar 4.23.

```

def stemming(komentar):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    x = []
    for word in komentar:
        komen = stemmer.stem(word)
        x.append(komen)
    komen_bersih = []
    komen_bersih = ' '.join(x)
    return komen_bersih
data['Stemming'] = data['Stopword'].apply(stemming)
data.head()

```

Gambar 4.23 Kode Penerapan Stemming

Setelah diimplementasikan *stemming* data komentar yang tadinya berupa token-token kemudian dijadikan kalimat kembali. Hasil pengimplementasian *stemming* dapat dilihat pada gambar 4.24.

Tokenize	Stopword	Stemming
[berhenti, pondok, ingin, kuliah, eh, malah, d...]	[berhenti, pondok, kuliah, disuruh, pondok]	henti pondok kuliah suruh pondok
[betul]	[]	
[tertawa, selamat, datang, di, ta, hukuman, se...]	[tertawa, selamat, hukuman, minggu]	tertawa selamat hukum minggu
[itu, segitu, ada, wajib, bayar, makan, lagi, ...]	[segitu, wajib, bayar, makan]	segitu wajib bayar makan
[belum, ada, bayar, katering, dan, penatu]	[bayar, katering, penatu]	bayar katering penatu

Gambar 4. 24 Hasil Penerapan *Stemming*

g). *Remove Comment*

Langkah selanjutnya yaitu proses penghapusan data komentar yang kosong atau tidak memiliki isi. Langkah ini diterapkan dengan menggunakan beberapa perintah kode yang berfungsi untuk

menghapus komentar yang kosong seperti yang ditampilkan pada gambar 2.25.

Remove Comment

```
def remove_comments(komentar):
    zero_length_comments = komentar[data["Stemming"].map(len) == 0]
    zero_length_comments_index = [ind for ind in zero_length_comments.index]
    data.drop(zero_length_comments_index, inplace = True)
    return data

data = remove_comments(data)
data.to_csv('data_clean.csv')
data.head(10)
```

Gambar 4. 25 Kode Penghapusan Komentar Kosong

Adapun hasil dari penghapusan komentar yang bernilai kosong adalah seperti gambar 2.26 berikut.

Slang1	Slang2	Tokenize	Stopword	Stemming
berhenti pondok ingin kuliah eh malah disuruh...	berhenti pondok ingin kuliah eh malah disuruh...	[berhenti, pondok, ingin, kuliah, eh, malah, d...]	[berhenti, pondok, kuliah, disuruh, pondok]	henti pondok kuliah suruh pondok
tertawa selamat datang di ta hukuman setiap mi...	tertawa selamat datang di ta hukuman setiap mi...	[tertawa, selamat, datang, di, ta, hukuman, se...]	[tertawa, selamat, hukuman, minggu]	tertawa selamat hukum minggu
itu segitu ada wajib bayar makan lagi apa ga	itu segitu ada wajib bayar makan lagi apa ga	[itu, segitu, ada, wajib, bayar, makan, lagi, ...]	[segitu, wajib, bayar, makan]	segitu wajib bayar makan
belum ada bayar katering dan penatu	belum ada bayar katering dan penatu	[belum, ada, bayar, katering, dan, penatu]	[bayar, katering, penatu]	bayar katering penatu
serius	serius	[serius]	[serius]	serius

Gambar 4.26 Hasil Kode Penghapusan Komentar Kosong

Setelah dilakukan *preprocessing*, data mengalami perubahan jumlah data dari sebelumnya 565 data menjadi 467 data. Secara spesifik jumlah data positif

dan negatif yang diperoleh setelah tahap *preprocessing* dapat ditunjukkan dengan gambar 4.27 berikut.

```
data['Label'].unique()
array(['negatif', 'positif'], dtype=object)

data.Label.value_counts()
negatif    279
positif    188
Name: Label, dtype: int64
```

Gambar 4. 27 Hasil Total Sentimen Pada Data
Komentar

Berdasarkan gambar diatas dapat diketahui bahwa terdapat komentar negatif sebanyak 297 komentar dan komentar positif sebanyak 188 komentar.

2. Processing

Adapun tahap selanjutnya yaitu *processing* yang mana data dilakukan *split validation* yang merupakan proses pembagian data menjadi data *testing* dan data *training*, kemudian dilakukan pembobotan kata dengan menggunakan TF-IDF, dan terakhir dilakukan *resampling* data dengan menggunakan teknik *oversampling*. Penelitian ini melakukan perbandingan antara *oversampling* menggunakan SMOTE dan *oversampling* menggunakan ADASYN. Berikut tahapan dari *processing* data:

a). Split Validation

Pada tahap ini akan dilakukan pembagian data menjadi data *training* (latih) dan data *testing* (uji) dengan perbandingan data latih 90% : 10% data uji. Adapun kode pembagian data menjadi data *training* dan data *testing* ditunjukkan oleh gambar 4.28.

```
2]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data['komentar'], data['label'], test_size= 0.1, random_state= 42)
X_train_dataframe = pd.DataFrame({'komentar': X_train, 'label': y_train})
X_train_dataframe.to_csv('X_Train_Dataset.csv')

print('X_train dataset: ', X_train.shape)
print('y_train dataset: ', y_train.shape)
print('X_test dataset: ', X_test.shape)
print('y_test dataset: ', y_test.shape)

X_train dataset: (420,)
y_train dataset: (420,)
X_test dataset: (47,)
y_test dataset: (47,)
```

Gambar 4. 28 Kode Pembagian Data *Training* Dan *Testing*

Berdasarkan pembagian data tersebut, data *training* memiliki 420 data sedangkan data *testing* memiliki 47 data.

b). Pembobotan TF-IDF

Pada tahap ini akan dilakukan pemberian bobot pada setiap kata yang ada pada data dengan menggunakan metode TF-IDF. TF-IDF merupakan suatu metode pembobotan kata yang dikenal baik dalam mengevaluasi pentingnya sebuah kata yang ada dalam dokumen (Khairunnisa et al., 2021).

Penerapan TF-IDF memerlukan *library scikitlearn* dan menggunakan modul *TfidfVectorizer*. Adapun kode pembobotan dengan menggunakan TF-IDF seperti pada gambar 4.29 berikut.

```
: from sklearn.feature_extraction.text import TfidfVectorizer
tf = TfidfVectorizer()
text_tf = tf.fit_transform(data['Komentar'])
X_train_tfidf = tf.transform(X_train)
X_test_tfidf = tf.transform(X_test)
X_train_df = pd.DataFrame(X_train_tfidf.toarray(), columns=tf.get_feature_names_out())
X_train_df.to_csv('X_Train_TFIDF.csv')
X_train_tfidf.toarray()
```

Gambar 4. 29 Kode Penerapan TF-IDF

Sehingga setelah tahap proses pembobotan dengan TF-IDF menghasilkan kalimat yang sudah

menjadi kumpulan *array* yang menjadi suatu matriks, dimana setiap baris mewakili setiap dokumen, sedangkan pada setiap kolom mewakili seluruh kata yang ada pada seluruh teks. Berikut hasil TF-IDF ditunjukkan pada gambar 4.30.

```
array([[0.      , 0.      , 0.      , ..., 0.      , 0.      ,
        0.      ],
       [0.      , 0.      , 0.      , ..., 0.      , 0.      ,
        0.      ],
       [0.      , 0.      , 0.84992942, ..., 0.      , 0.      ,
        0.      ],
       ...,
       [0.      , 0.      , 0.      , ..., 0.      , 0.      ,
        0.      ],
       [0.      , 0.      , 0.      , ..., 0.32924493, 0.      ,
        0.      ],
       [0.      , 0.      , 0.      , ..., 0.11796076, 0.      ,
        0.      ]])
```

Gambar 4. 30 Hasil Proses TF-IDF

c). *Resampling*

Setelah melalui tahap pembagian data menjadi data *training* dan data *testing*, selanjutnya data melalui proses *resampling*. Tujuan *resampling* adalah memodifikasi dataset sehingga lebih proporsional atau lebih representatif. Ketidakseimbangan data akan terjadi apabila jumlah objek di suatu kelas data memiliki kuantitas yang lebih tinggi dibandingkan dengan kelas lain, dimana kelas data yang objeknya lebih banyak disebut kelas mayor sedangkan yang lain disebut minor. Pengolahan algoritma yang tidak mempertimbangkan ketidakseimbangan data cenderung

menitikberatkan kelas mayor dan bukan kelas minor, oleh sebab itu diperlukan *oversampling* untuk memperbanyak pengamatan secara acak dengan menambah jumlah data kelas minor (data buatan) agar setara dengan kelas mayor (Yerik Afrianto Singgalen, 2023). Dalam implementasinya, dibutuhkan *library imblearn* tanpa memodifikasi parameter apapun atau menggunakan *default setting* (Nurhopipah & Magnolia, 2022).

Pada penelitian ini dilakukan perbandingan dua teknik *oversampling* yaitu *oversampling* dengan menggunakan SMOTE dan *oversampling* dengan menggunakan ADASYN. Penerapan *oversampling* memerlukan modul SMOTE yang disediakan oleh *library imblearn*. Berikut kode penerapan *oversampling* dengan menggunakan SMOTE.

Oversampling SMOTE

```
1: from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state = 42)
X_train_smote, y_train_smote, = smote.fit_resample(X_train_tfidf, y_train.ravel())

smote_df = pd.DataFrame(X_train_smote.toarray(), columns=tf.get_feature_names_out(), index=y_train_smote)
smote_df.to_csv('hasil_smote.csv')

print("Sebelum oversampling, jumlah label 'positif': {}".format(sum(y_train == 'positif')))
print("Sebelum oversampling, jumlah label 'negatif': {}".format(sum(y_train == 'negatif')))

print("Setelah oversampling, bentuk train_X: {}".format(X_train_smote.shape))
print("Setelah oversampling, bentuk train_y: {}".format(y_train_smote.shape))

print("Setelah oversampling, jumlah label 'positif': {}".format(sum(y_train_smote == 'positif')))
print("Setelah oversampling, jumlah label 'negatif': {}".format(sum(y_train_smote == 'negatif')))
```

Gambar 4.31 Kode Penerapan *Oversampling* SMOTE

Adapun penerapan *oversampling* dengan menggunakan ADASYN juga memerlukan modul ADASYN yang disediakan oleh *library imblearn*.

Oversampling ADASYN

```
from imblearn.over_sampling import ADASYN
adasyn = ADASYN(random_state = 42)
X_train_adasyn, y_train_adasyn, = adasyn.fit_resample(X_train_tfidf, y_train)

adasyn_df = pd.DataFrame(X_train_adasyn.toarray(), columns=tf.get_feature_names_out(), index=y_train_adasyn)
adasyn_df.to_csv('hasil_adasyn.csv')

print("Sebelum oversampling, jumlah label 'positif': {}".format(sum(y_train == 'positif')))
print("Sebelum oversampling, jumlah label 'negatif': {}".format(sum(y_train == 'negatif')))

print("Setelah oversampling, bentuk train_X: {}".format(X_train_adasyn.shape))
print("Setelah oversampling, bentuk train_y: {}".format(y_train_adasyn.shape))

print("Setelah oversampling, jumlah label 'positif': {}".format(sum(y_train_adasyn == 'positif')))
print("Setelah oversampling, jumlah label 'negatif': {}".format(sum(y_train_adasyn == 'negatif')))
```

Gambar 4. 32 Kode Penerapan *Oversampling* ADASYN

Setelah diterapkan *oversampling* dengan SMOTE data yang berlabel positif sebelumnya bernilai 170 data menjadi 250 data, sedangkan data yang berlabel negatif tetap menjadi 250. Artinya bahwa *oversampling* membuat sampel baru dari kelas minoritas sehingga memiliki jumlah data seimbang. Gambar 4.33 merupakan hasil penerapan *oversampling* dengan SMOTE.

```
Sebelum oversampling, jumlah label 'positif': 170
Sebelum oversampling, jumlah label 'negatif': 250

Setelah oversampling, bentuk train_X: (500, 756)
Setelah oversampling, bentuk train_y: (500,)

Setelah oversampling, jumlah label 'positif': 250
Setelah oversampling, jumlah label 'negatif': 250
```

Gambar 4. 33 Hasil Implementasi *Oversampling* SMOTE

Adapun penerapan *oversampling* dengan ADASYN yaitu menghasilkan sampel baru pada kelas positif yang sebelumnya berjumlah 170 data menjadi 246 data. Artinya bahwa *oversampling* ADASYN membentuk sampel baru dengan jumlah mendekati kelas mayoritasnya yaitu kelas negatif yang berjumlah 250 data. Gambar 4.34 merupakan hasil dari penerapan *oversampling* dengan ADASYN.

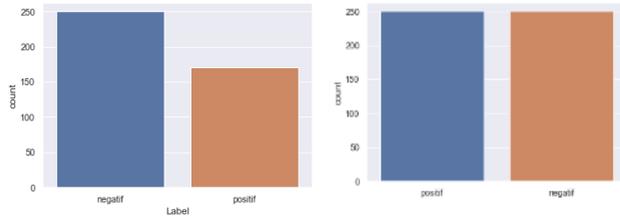
```
Sebelum oversampling, jumlah label 'positif': 170
Sebelum oversampling, jumlah label 'negatif': 250

Setelah oversampling, bentuk train_X: (496, 756)
Setelah oversampling, bentuk train_y: (496,)

Setelah oversampling, jumlah label 'positif': 246
Setelah oversampling, jumlah label 'negatif': 250
```

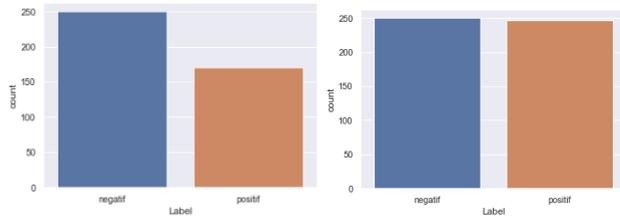
Gambar 4. 34 Hasil Implementasi *Oversampling* ADASYN

Jika divisualisasikan menggunakan *seaborn*, data sebelum *oversampling* dengan SMOTE dan setelah diterapkan SMOTE dapat dilihat pada gambar 4.35 berikut ini.



Gambar 4. 35 Perbandingan Data Sebelum Dan Setelah *Oversampling* SMOTE

Adapun visualisasi data sebelum dan setelah diterapkan *oversampling* dengan ADASYN adalah pada gambar 4.36 berikut.



Gambar 4. 36 Perbandingan Data Sebelum Dan Setelah *Oversampling* ADASYN

D. *Modeling*

Setelah melalui tahapan *data preparation*, selanjutnya data akan masuk kedalam tahap *modeling*. Pada penelitian ini pemodelan dilakukan dengan menggunakan metode *Naïve Bayes Classifier* untuk memperoleh probabilitas apakah termasuk kedalam kelas positif atau kelas negatif. Pengklasifikasian *naïve bayes* dilakukan dengan memasukan data *training* sebesar 90%

kemudian diuji ketepatannya dengan menggunakan data *testing*. Pengklasifikasian dilakukan dengan memanfaatkan *library sklearn* dengan memanggil beberapa modul yang tersedia didalamnya seperti *MultinomialNB*, *accuracy_score*, *precision_score*, *recall_score*, *f1_score*, *classification_report* dan *confusion matrix*. Kode pemanggilan modul dari *library sklearn* dapat dilihat pada gambar 4.37 berikut.

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

Gambar 4. 37 Kode Pemanggilan Modul *Library Sklearn*

Selanjutnya pengklasifikasian menggunakan *Naive Bayes* dengan oversampling SMOTE seperti pada gambar 4.38 dibawah ini.

```
clf_sm = MultinomialNB().fit(X_train_smote, y_train_smote)
predicted_sm = clf_sm.predict(X_test_tfidf)

sentimen_smote_df = pd.DataFrame({'komentar': X_test, 'label': y_test, 'prediksi': predicted_sm})
sentimen_smote_df.to_csv('Sentimen_SMOTE.csv', index=False)

print("MultinomialNB Accuracy:", accuracy_score(y_test,predicted_sm))
print("MultinomialNB Precision:", precision_score(y_test,predicted_sm, average='weighted'))
print("MultinomialNB Recall:", recall_score(y_test, predicted_sm, average='weighted'))
print("MultinomialNB f1_score:", f1_score(y_test,predicted_sm, average='weighted'))
```

Gambar 4. 38 Kode Proses Pengklasifikasian *Naive Bayes*
Dengan SMOTE

Sedangkan kode pengklasifikasian menggunakan *Naive Bayes* dengan oversampling ADASYN seperti pada gambar 4.39 dibawah ini.

```
clf_ad = MultinomialNB().fit(X_train_adasyn, y_train_adasyn)
predicted_ad = clf_ad.predict(X_test_tfidf)

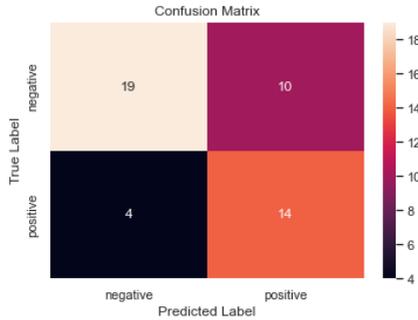
hasil_df = pd.DataFrame({'komentar': X_test, 'label': y_test, 'prediksi': predicted_ad})
hasil_df.to_csv('Sentimen_ADASYN.csv', index=False)

print("MultinomialNB Accuracy:", accuracy_score(y_test,predicted_ad))
print("MultinomialNB Precision:", precision_score(y_test,predicted_ad, average='weighted'))
print("MultinomialNB Recall:", recall_score(y_test, predicted_ad, average='weighted'))
print("MultinomialNB f1_score:", f1_score(y_test,predicted_ad, average='weighted'))
```

Gambar 4. 39 Kode Proses Pengklasifikasian *Naive Bayes* Dengan ADASYN

E. Evaluation

Selanjutnya dilakukan tahap evaluasi yang digunakan untuk menilai kesesuaian model dalam pengklasifikasian. Pada proses ini akan ditampilkan perbandingan hasil performa model dari data yang telah dilakukan *oversampling* dengan menggunakan SMOTE dan ADASYN. Performa pada model tersebut didapatkan dengan melakukan perhitungan *confusion matrix*. Nilai perhitungan performa diukur dengan menghitung nilai *accuracy*, *presicion*, *recall* dan *f1 score*. Berikut *confusion matrix* yang diperoleh dari penerapan algoritma *Naive Bayes* dengan *oversampling* SMOTE yang ditunjukkan pada gambar 4.40:



Gambar 4. 40 *Confusion Matrix* Data SMOTE

Confusion matrix diatas digunakan untuk melakukan perhitungan nilai *accuracy*, *precision*, *recall*, dan *f1_score*. Berikut perhitungan manualnya:

$$\begin{aligned}
 Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \times 100 \% = \frac{14+19}{14+19+10+4} \times 100 \% \\
 &= 0,70 \times 100 \% = \mathbf{70 \%}
 \end{aligned}$$

- Kelas Positif

$$\begin{aligned}
 Precision &= \frac{TP}{TP+FP} \times 100 \% = \frac{14}{14+10} \times 100 \% \\
 &= 0,58 \times 100 \% \\
 &= \mathbf{58\%}
 \end{aligned}$$

$$\begin{aligned}
 Recall &= \frac{TP}{TP+FN} \times 100 \% = \frac{14}{14+4} \times 100 \% \\
 &= 0,777 \times 100 \% \\
 &= \mathbf{78\%}
 \end{aligned}$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100 \%$$

$$\begin{aligned}
&= 2 \times \frac{0,58 \times 0,78}{0,58 + 0,78} \times 100 \% = 2 \times \frac{0,4524}{1,36} \times 100 \% \\
&= 0,67 \times 100 \% \\
&= \mathbf{67\%}
\end{aligned}$$

- Kelas Negatif

$$\begin{aligned}
\text{Presisi} &= \frac{TN}{TN + FN} \times 100 \% = \frac{19}{19 + 4} \times 100 \% \\
&= 0,83 \times 100 \% \\
&= \mathbf{83\%}
\end{aligned}$$

$$\begin{aligned}
\text{Recall} &= \frac{TN}{TN + FP} \times 100 \% = \frac{19}{19 + 10} \times 100 \% \\
&= 0,66 \times 100 \% \\
&= \mathbf{66\%}
\end{aligned}$$

$$\begin{aligned}
\text{F1-Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100 \% \\
&= 2 \times \frac{0,83 \times 0,66}{0,83 + 0,66} \times 100 \% = 2 \times \frac{0,5478}{1,49} \times 100 \% \\
&= 0,73 \times 100 \% \\
&= \mathbf{73 \%}
\end{aligned}$$

Selanjutnya menentukan perhitungan *weighted average* yang merupakan rata-rata nilai presisi, *recall* dan *f1_score* dari masing-masing kelas dengan memberikan bobot kepada setiap kelas berdasarkan jumlah contoh dalam kelas tersebut. Jumlah contoh pada kelas positif yakni 18 sedangkan pada kelas negatif 29. Berikut perhitungan manualnya:

$$\text{Bobot Kelas Positif} = \frac{18}{(18+29)} = \mathbf{0,39}$$

$$\text{Bobot Kelas Negatif} = \frac{29}{(18+29)} = \mathbf{0,62}$$

$$\begin{aligned} \text{Weighted Avg Precision} &= (0,58 \times 0,39) + (0,83 \times 0,62) \times 100\% \\ &= 0,22 + 0,51 \times 100\% = 0,73 \times 100\% \\ &= \mathbf{73 \%} \end{aligned}$$

$$\begin{aligned} \text{Weighted Avg Recall} &= (0,78 \times 0,39) + (0,66 \times 0,62) \times 100\% \\ &= 0,30 + 0,40 \times 100\% = 0,70 \times 100\% \\ &= \mathbf{70 \%} \end{aligned}$$

$$\begin{aligned} \text{Weighted Avg F1-Score} &= (0,67 \times 0,39) + (0,73 \times 0,62) \times 100\% \\ &= 0,26 + 0,45 \times 100\% = 0,71 \times 100\% \\ &= \mathbf{71\%} \end{aligned}$$

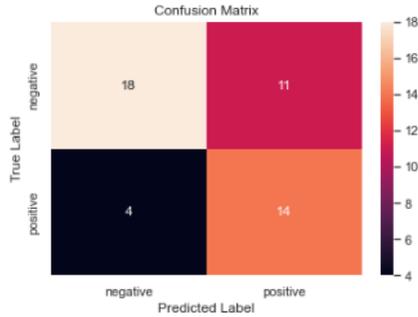
Berdasarkan perhitungan diatas diketahui performa dari algoritma *Naïve Bayes* dengan *oversampling* SMOTE memperoleh akurasi 70%, presisi 73 %, *recall* 70%, *F1-score* 71%. Adapun hasil penghitungan yang dilakukan sistem adalah seperti pada gambar 4.41 berikut:

	precision	recall	f1-score	support
negatif	0.83	0.66	0.73	29
positif	0.58	0.78	0.67	18
accuracy			0.70	47
macro avg	0.70	0.72	0.70	47
weighted avg	0.73	0.70	0.71	47

Gambar 4. 41 Nilai Performa Dari Pemodelan NBC Dengan SMOTE

Berdasarkan hasil tersebut dapat diketahui bahwa nilai *precision* atau tingkat keberhasilan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna pada nilai negatif sebesar 83%, sedangkan pada nilai positif sebesar 58%. sehingga dari angka tersebut dapat diartikan bahwa proporsi label yang diprediksi dengan negatif lebih tinggi dibandingkan dengan label positif. Nilai *recall* pada nilai negatif sebesar 66%, sedangkan pada nilai positif sebesar 78%. Artinya nilai *recall* atau tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi yang bernilai negatif lebih besar dari label positif. Untuk nilai *f1-score* dapat dihitung sebesar 72% pada nilai negatif dan 67% pada nilai positif. Sehingga diperoleh total keseluruhan dari nilai *precision*, *recall* serta *f1 score* pada gambar 4.46 didapatkan nilai *precision* sebesar 73%, nilai *recall* sebesar 70% dan nilai *f1 score* sebesar 71%.

Selanjutnya perhitungan performa algoritma *Naïve Bayes* dengan *oversampling* ADASYN. Perhitungan nilai akurasi, presisi, *recall*, dan *f1-score* berdasarkan perhitungan dari *confusion matrix*. Gambar 4.42 berikut *confusion matrix* yang diperoleh dari penerapan algoritma *Naïve Bayes* dengan *oversampling* ADASYN:



Gambar 4. 42 *Confusion Matrix* Data ADASYN

Confusion matrix diatas digunakan untuk melakukan perhitungan nilai *accuracy*, *precision*, *recall*, dan *f1_score*. Berikut perhitungan manualnya:

$$\begin{aligned}
 Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \times 100 \% = \frac{14+18}{14+18+11+4} \times 100 \% \\
 &= 0,68 \times 100 \% \\
 &= \mathbf{68 \%}
 \end{aligned}$$

- Kelas Positif

$$\begin{aligned}
 Presisi &= \frac{TP}{TP+FP} \times 100 \% = \frac{14}{14+11} \times 100 \% \\
 &= 0,56 \times 100 \% \\
 &= \mathbf{56\%}
 \end{aligned}$$

$$\begin{aligned}
 Recall &= \frac{TP}{TP+FN} \times 100 \% = \frac{14}{14+4} \times 100 \% \\
 &= 0,777 \times 100 \% \\
 &= \mathbf{78\%}
 \end{aligned}$$

$$\begin{aligned}
 F1\text{-Score} &= 2 \times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} \times 100 \% \\
 &= 2 \times \frac{0,56 \times 0,78}{0,56 + 0,78} \times 100 \% = 2 \times \frac{0,4368}{1,34} \times 100 \% \\
 &= 0,65 \times 100 \% \\
 &= \mathbf{65\%}
 \end{aligned}$$

- Kelas Negatif

$$\begin{aligned}
 \textit{Precision} &= \frac{TN}{TN + FN} \times 100 \% = \frac{18}{18 + 4} \times 100 \% \\
 &= 0,82 \times 100 \% \\
 &= \mathbf{82\%}
 \end{aligned}$$

$$\begin{aligned}
 \textit{Recall} &= \frac{TN}{TN + FP} \times 100 \% = \frac{18}{18 + 11} \times 100 \% \\
 &= 0,62 \times 100 \% \\
 &= \mathbf{62\%}
 \end{aligned}$$

$$\begin{aligned}
 F1\text{-Score} &= 2 \times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} \times 100 \% \\
 &= 2 \times \frac{0,82 \times 0,62}{0,82 + 0,62} \times 100 \% = 2 \times \frac{0,5084}{1,44} \times 100 \% \\
 &= 0,71 \times 100 \% \\
 &= \mathbf{71 \%}
 \end{aligned}$$

Selanjutnya menentukan perhitungan *weighted average* yang merupakan rata-rata nilai *precision*, *recall* dan *f1_score* dari masing-masing kelas dengan memberikan bobot kepada setiap kelas berdasarkan jumlah contoh dalam kelas tersebut. Jumlah

contoh pada kelas positif yakni 18 sedangkan pada kelas negatif 29. Berikut perhitungan manualnya:

$$\text{Bobot Kelas Positif} = \frac{18}{(18+29)} = \mathbf{0,39}$$

$$\text{Bobot Kelas Negatif} = \frac{29}{(18+29)} = \mathbf{0,62}$$

$$\begin{aligned} \text{Weighted Avg Precision} &= (0,56 \times 0,39) + (0,82 \times 0,62) \times 100\% \\ &= 0,22 + 0,50 \times 100\% = 0,72 \times 100\% \\ &= \mathbf{72 \%} \end{aligned}$$

$$\begin{aligned} \text{Weighted Avg Recall} &= (0,78 \times 0,39) + (0,62 \times 0,62) \times 100\% \\ &= 0,30 + 0,38 \times 100\% = 0,68 \times 100\% \\ &= \mathbf{68 \%} \end{aligned}$$

$$\begin{aligned} \text{Weighted Avg F1-Score} &= (0,64 \times 0,39) + (0,71 \times 0,62) \times 100\% \\ &= 0,24 + 0,44 \times 100\% = 0,68 \times 100\% \\ &= \mathbf{68\%} \end{aligned}$$

Berdasarkan perhitungan diatas diketahui performa dari algoritma *Naïve Bayes* dengan *oversampling* ADASYN memperoleh akurasi 68%, presisi 72 %, *recall* 68%, F1-score 68%. Adapun hasil penghitungan yang dilakukan sistem adalah sebagai berikut:

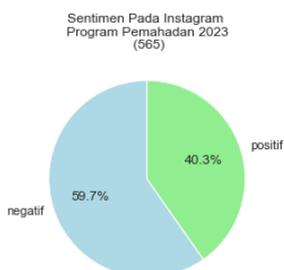
	precision	recall	f1-score	support
negatif	0.82	0.62	0.71	29
positif	0.56	0.78	0.65	18
accuracy			0.68	47
macro avg	0.69	0.70	0.68	47
weighted avg	0.72	0.68	0.68	47

Gambar 4. 43 Nilai Performa Dari Pemodelan NBC Dengan ADASYN

Berdasarkan hasil tersebut dapat diketahui bahwa nilai *precision* atau tingkat keberhasilan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna pada nilai negatif sebesar 82%, sedangkan pada nilai positif sebesar 56%. sehingga dari angka tersebut dapat diartikan bahwa proporsi label yang diprediksi dengan negatif lebih tinggi dibandingkan dengan label positif. Nilai *recall* pada nilai negatif sebesar 62%, sedangkan pada nilai positif sebesar 78%. Artinya nilai *recall* atau tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi yang bernilai positif lebih besar dari label negatif. Untuk nilai *f1-score* dapat dihitung sebesar 71% pada nilai negatif dan 65% pada nilai positif . Sehingga diperoleh total keseluruhan dari nilai *precision*, *recall* serta *f1 score* pada gambar 4.47 didapatkan nilai *precision* sebesar 72%, nilai *recall* sebesar 68% dan nilai *f1 score* sebesar 68%.

F. Deployment

Tahap *deployment* menjadi tahap terakhir yaitu dengan memvisualisasikan hasil dari analisis sentimen kedalam diagram lingkaran dan *wordcloud*. *Wordcloud* akan digunakan untuk memvisualisasikan hasil analisis klasifikasi. Tujuan dari visualisasi tersebut untuk mengetahui jumlah sentimen dan kata yang sering dibicarakan pada topik terkait Kebijakan Program Pema'hadan UIN Walisongo Tahun 2023. Maka dari itu dengan memvisualisasikannya dapat diperoleh informasi yang ditampilkan dalam bentuk gambar.



Gambar 4. 44 Persentase Pada Sentimen

Gambar 4.44 menunjukkan persentase dari jumlah data bernilai positif, dan negatif. Berdasarkan diagram tersebut dapat diketahui bahwa data yang bernilai positif memiliki nilai sebesar 40.3 %. Adapun data bernilai negatif memiliki persentase yang lebih tinggi yaitu sebesar 59.7 %. Sehingga dari angka tersebut diperoleh bahwa tanggapan mengenai Kebijakan Program

“universitas islam”, “uang kuliah”, “tertawa”, “kuliah tunggal”, “kartu Indonesia”, “kampus”, “mahasiswa”, dan “indonesia pintar” merupakan 10 kata yang paling sering dibicarakan dari pengguna media sosial Instagram.

Kemudian peneliti juga ingin mengetahui *wordcloud* dari sentimen positif dan negatif. Berikut pada gambar 4.46 menampilkan *wordcloud* dari sentimen positif.



Gambar 4. 46 *Wordcloud* Pada Sentimen Positif

Pada gambar 4.46 dapat disimpulkan, yaitu informasi pada kata yang sering muncul dalam sentimen positif ialah kata ‘pondok’, ‘universitas islam’, ‘islam negeri’, ‘tertawa’, ‘wajib’, ‘bayar’, ‘mahasiswa’, ‘kartu indonesia’, ‘masuk’, ‘indonesia pintar’ dan lain-lainnya. Selanjutnya peneliti akan menampilkan *wordcloud* dari sentimen negatif, yang mana kata yang tergolong sentimen negatif perlu diperhatikan dan dievaluasi untuk mengoptimalkan kebijakan program pema’hadan UIN Walisongo Semarang.



Gambar 4.46 *Wordcloud* Pada Sentimen Negatif

Sedangkan pada kata yang sering muncul dalam sentimen negatif ialah kata ‘pondok’, ‘uang kuliah’, ‘universitas islam’, ‘kuliah tunggal’, ‘islam negeri’, ‘kampus’, ‘bayar’, ‘biaya’, ‘bayar’, ‘indonesia pintar’, dan lainnya. Artinya, kata-kata tersebut menjadi kata yang sering dibicarakan oleh pengguna media sosial Instagram karena kata tersebut dicetak lebih besar dari pada kata-kata lainnya.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang dilaksanakan, dapat diambil kesimpulan bahwa :

1. Data yang didapat dari komentar Instagram sebanyak 565 data, dianalisis dengan algoritma *Naive Bayes Classifier* menggunakan metode CRISP-DM. Adapun urutan proses CRISP-DM yaitu *business understanding, data understanding, data preparation, modeling, evaluation, deployment*. Algoritma *Naive Bayes* diterapkan pada proses *modeling* dan sekaligus dilakukan tahapan *resampling* data dengan teknik *oversampling* untuk menangani ketidakseimbangan data. Kemudian dilakukan perbandingan antara algoritma *Naive Bayes* menggunakan *oversampling* SMOTE dengan algoritma *Naive Bayes* menggunakan *oversampling* ADASYN. Adapun proporsi pembagian data pada penelitian ini sebesar 90% data latih dan 10% data uji dari keseluruhan jumlah data yang dipilih secara acak. Hasil klasifikasi yang diberikan berupa kelas dengan sentimen positif, dan negatif.
2. Dari penelitian ini menghasilkan klasifikasi sebanyak 188 komentar bersentimen positif dan 279 komentar sentimen negatif. Sehingga diketahui sentimen negatif

memiliki nilai persentase tertinggi sebesar 59.7%, sedangkan pada sentimen positif sebesar 40.3%. Adapun performa yang diberikan algoritma Naive Bayes dengan menggunakan SMOTE mendapatkan nilai akurasi sebesar 70%, *precision* sebesar 73%, *recall* sebesar 70%, serta *f1 score* sebesar 71%, dan *Naive Bayes* dengan menggunakan ADASYN mendapatkan nilai akurasi sebesar 68%, *precision* sebesar 72%, *recall* sebesar 68%, *f1-score* sebesar 68%. Sehingga dapat dikatakan bahwa hasil penggunaan SMOTE dapat meningkatkan performa *Naive Bayes* lebih tinggi dibandingkan dengan ADASYN dalam analisis sentimen.

B. Saran

Berdasarkan penelitian yang dilaksanakan, penulis berharap kepada peneliti selanjutnya untuk dapat dikembangkan dan terdapat beberapa saran-saran, yaitu:

1. Algoritma yang berbeda dapat diterapkan agar dapat dilakukan perbandingan dalam mencari hasil klasifikasi yang terbaik. Algoritma lainnya seperti *Support Vector Machine* (SVM), K-NN, *Decision Tree*, *Random Forest* dan lain sebagainya.

2. Data dalam penelitian ini berasal dari komentar Instagram, disarankan pada penelitian berikutnya dapat mengambil data dari media sosial lainnya seperti Youtube, Tik Tok, Google Play Store, Facebook, dan lain sebagainya.
3. Penambahan koleksi kamus pada kata yang tidak baku/gaul, karena pada media sosial banyak komentar yang berisikan bahasa yang kurang baku.
4. Menggunakan metode *resampling* lainnya seperti ROS, RUS, SMOTE-ENN, *NearMiss*, *Cluster Centroid*, dan lainnya agar dapat membandingkan hasilnya.

DAFTAR PUSTAKA

- Afdhal, I., Kurniawan, R., Iskandar, I., Salambue, R., Budianita, E., & Syafrina, F. (2022). Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia. *Jurnal Nasional Komputasi Dan Teknologi Informasi*, 5(1), 122–130. <http://ojs.serambimekkah.ac.id/jnknti/article/view/4004/pdf>
- Afrianto Singgalen, Y., Jend Sudirman No, J., Semanggi, K., Setiabudi, K., Jakarta Selatan, K., & Khusus, D. (2023). Analisis Sentimen dan Sistem Pendukung Keputusan Menginap di Hotel Menggunakan Metode CRISP-DM dan SAW. *Journal of Information System Research*, 4(4), 1343–1353. <https://doi.org/10.47065/josh.v4i4.3917>
- Akhir, M. T., Syarat, M., Memperoleh, G., Sarjana, G., Satu, S., & Informasi, T. (2022). Analisis Sentimen Kenaikan Harga Bahan Bakar Minyak (BBM) Berdasarkan Respon Pengguna Media Sosial Twitter Di Indonesia Menggunakan Metode Naive Bayes. *SKRIPSI*.
- Ardiani, L., Sujaini, H., & Tursina, T. (2020). Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, 8(2), 183. <https://doi.org/10.26418/justin.v8i2.36776>
- Ayu Purnamasari S, Muhammad Amin, L. J. L. A. R. (2022). Education and Learning Journal. *Universitas Muslim Indonesia*, 1(January), 106–113. <https://jurnal.fai.umi.ac.id/index.php/eljour/>
- Azizah, H. N. (2023). Visualisasi Analisis Sentimen Siberbullying Pada Post Instagram Menggunakan Orange Data Mining. *IJCSR: The Indonesian Journal of Computer Science Research*, 1(1), 42–48. <https://subset.id/index.php/IJCSR>
- Chely Aulia Misrun, Haerani, E., Fikry, M., & Budianita, E. (2023). Analisis sentimen komentar youtube terhadap Anies Baswedan sebagai bakal calon presiden 2024 menggunakan

- metode naive bayes classifier. *Jurnal CoSciTech (Computer Science and Information Technology)*, 4(1), 207–215. <https://doi.org/10.37859/coscitech.v4i1.4790>
- Dedy, I. P., Darmawan, W., Pradnyana, G. A., Bagus, I., & Pascima, N. (2023). *Optimasi Parameter Support Vector Machine Dengan Algoritma Genetika Untuk Analisis Sentimen Pada Media Sosial Instagram*. 6(1), 58–67.
- Febilianingtyas, A., & Febriana, P. (n.d.). *Analysis of Instagram Followers @ sparklingsurabaya About City Branding (Analysis Receptions of Overseas Student in Surabaya) [Analisis Followers Instagram @ sparklingsurabaya Tentang City Branding (Analisis Resepsi Mahasiswa Rantau di Surabaya)]*. 1–9.
- Febriyani, E., & Februariyanti, H. (2022). *Analisis Sentimen Terhadap Program Kampus Merdeka Menggunakan Algoritma Naive Bayes Classifier Di Twitter*. 17(1), 25–38.
- Fitrianti, I., Voutama, A., & Umaidah, Y. (2023). Clustering Film Populer pada Aplikasi Netflix dengan Menggunakan Algoritma K-Means dan Metode CRISP-DM. *Jurnal Teknologi Sistem ...*, 4(2), 301–311. <https://jurnal.mdp.ac.id/index.php/jtsi/article/view/4929%0Ahttps://jurnal.mdp.ac.id/index.php/jtsi/article/download/4929/1546>
- Flores, V. A., Jasa, L., & Linawati, L. (2020). Analisis Sentimen untuk Mengetahui Kelemahan dan Kelebihan Pesaing Bisnis Rumah Makan Berdasarkan Komentar Positif dan Negatif di Instagram. *Majalah Ilmiah Teknologi Elektro*, 19(1), 49. <https://doi.org/10.24843/mite.2020.v19i01.p07>
- Harpizon, H. A. R., Kurniawan, R., Iwan Iskandar, Salambue, R., Budianita, E., & Syafria, F. (2022). Analisis Sentimen Komentar Di YouTube Tentang Ceramah Ustadz Abdul Somad Menggunakan Algoritma Naïve Bayes. *JNKTI (Jurnal Nasional Komputasi Dan Teknologi Informasi)*, 5(1), 131–140. <http://repository.uin-suska.ac.id/59746/>
- Khairunnisa, S., Adiwijaya, A., & Faraby, S. Al. (2021). Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi

- Kasus Pandemi COVID-19). *Jurnal Media Informatika Budidarma*, 5(2), 406.
<https://doi.org/10.30865/mib.v5i2.2835>
- Kurniawan, B., Suwarisman, A., Afriyanti, I., Wahyudi, A., & Saputra, D. D. (2023). Analisis Sentimen Complain dan Bukan Complain pada Twitter Telkomsel dengan SMOTE dan Naïve Bayes. *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, 7(1), 106–113.
<https://doi.org/10.35870/jtik.v7i1.691>
- Luthfiyyah, I. Q., Sari, B. N., & Ridwan, T. (2024). *Analisis Sentimen Mahasiswa Terhadap Kurikulum Literasi Digital di Universitas Singaperbangsa Karawang Menggunakan Naïve Bayes*. 19(1), 122–136.
- Mustofa, H., & Mahfudh, A. A. (2019). Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes. *Walisongo Journal of Information Technology*, 1(1), 1.
<https://doi.org/10.21580/wjit.2019.1.1.3915>
- Neightbors, K. (2023). *Analisis Sentimen Terhadap Pemandangan Ibu Kota Negara Menggunakan Algoritma Naive Bayes Classifier*. 7(4), 1306–1314.
<https://doi.org/10.30865/mib.v7i3.6523>
- Ningtyas, A. A., Solichin, A., & Pradana, R. (2023). Analisis Sentimen Komentar Youtube Tentang Prediksi Resesi Ekonomi Tahun 2023 Menggunakan Algoritme Naïve Bayes. *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, 20(1), 9–16.
- Nurhopipah, A., & Magnolia, C. (2022). Perbandingan Metode Resampling Pada Imbalanced Dataset Untuk Klasifikasi Komentar Program Mbkm. *Jurnal Publikasi Ilmu Komputer Dan Multimedia*, 1(2), 9–22.
- Rahayu, I. P., Fauzi, A., & Indra, J. (2022). *Analisis Sentimen Terhadap Program Kampus Merdeka Menggunakan Naive Bayes Dan Support Vector Machine*. 4, 296–301.
<https://doi.org/10.30865/json.v4i2.5381>
- Sentimen, A., Maskapai, P., Husada, H. C., & Paramita, A. S. (2021). *Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector*

- Machine (SVM) Sentiment Analysis of Airline on Twitter Platform Using Support Vector Machine (SVM) Algorithm. 10(1), 18–26. <https://doi.org/10.34148/teknika.v10i1.311>*
- Singgalen, Y A. (2023). ... Metode CRISP-DM dalam Klasifikasi Data Ulasan Pengunjung Destinasi Danau Toba Menggunakan Algoritma Naïve Bayes Classifier (NBC) dan Decision Tree (DT). *Jurnal Media Informatika Budidarma*, 7, 1551–1562. <https://doi.org/10.30865/mib.v7i3.6461>
- Singgalen, Yerik Afrianto. (2023). *Penerapan CRISP-DM dalam Klasifikasi Sentimen dan Analisis Perilaku Pembelian Layanan Akomodasi Hotel Berbasis Algoritma Decision Tree (DT)*. 5, 237–248. <https://doi.org/10.30865/json.v5i2.7081>
- Susianti, I., Ningsih, S. S., Haris, M. Al, Utami, T. W., Universitas, M., Semarang, M., Universitas, M., Semarang, M., Universitas, D., Semarang, M., Universitas, D., & Semarang, M. (2020). *Prosiding Seminar Edusainstech FMIPA UNIMUS 2020 ISBN: 978-602-5614-35-4 ANALISIS SENTIMEN PADA TWITTER TERKAIT NEW NORMAL Prosiding Seminar Edusainstech FMIPA UNIMUS 2020 ISBN: 978-602-5614-35-4*. 354–363.
- Tuhuteru, H., Iriani, A., Informasi, M. S., Informasi, F. T., Kristen, U., Wacana, S., Salatiga, K., & Bersama, P. H. (2018). *Analisis Sentimen Perusahaan Listrik Negara Cabang Ambon Menggunakan Metode Support Vector Machine dan Naive Bayes Classifier. 03(03)*, 394–401. <https://doi.org/10.30591/jpit.v3i3.977>
- Turmudi Zy, A., Adji Ardiansyah, L., & Maulana, D. (2021). Implementasi Algoritma Naïve Bayes Dalam Mendiagnosa Penyakit Angin Duduk. *Jurnal Pelita Teknologi*, 16(1), 52–65.
- Yuniarti, W. D., Faiz, A. N., & Setiawan, B. (2020). Identifikasi Potensi Keberhasilan Studi Menggunakan Naïve Bayes Classifier. *Walisongo Journal of Information Technology*, 2(1), 1. <https://doi.org/10.21580/wjit.2020.2.1.5204>
- Zhafira, D. F., Rahayudi, B., Brawijaya, U., & Korespondensi, P. (2021). *ANALISIS SENTIMEN KEBIJAKAN KAMPUS*

MERDEKA MENGGUNAKAN NAIVE BAYES DAN PEMBOBOTAN TF-IDF SENTIMENT ANALYSIS OF KAMPUS MERDEKA POLICY USING NAIVE BAYES AND TF-IDF TERM WEIGHTING BASED ON YOUTUBE. 2(1), 55–63.

Zidan, Muhammad. (2022). Analisis Sentimen Kenaikan Harga Bahan Bakar Minyak (BBM) Berdasarkan Respon Pengguna Media Sosial Twitter Di Indonesia Menggunakan Metode Naive Bayes. *SKRIPSI*.

LAMPIRAN

Lampiran 1. Contoh Hasil Eksport Data Instagram

Username	Komentar
user001	mandek mondok pengen kuliah, eh malah dikon mondok meneh anjirrr
user002	@imchoice19__ betuuulll
user003	Akwoksowk welcome to ta'dziran tiap minggu
user004	Itu segitu ada wajib bayar makan lagi apa ga?
user005	@heni_fatmaa belum.. ada bayar katering dan laundry
user006	@marco_marconah serius?
user007	@marco_marconah sekitar berapa?
user008	@wahyunugro hosp katering nya 450 k/ bulan terus laundry nya per kg 7 k
user009	@wahyunugro hosp Ya gak tau wkwkwk heheheh bisa2 intinya sabar aku juga pemaksaan kok
user010	@nittcollage gmn komentarnya soal makanannya?
user011	@marco_marconah ya gitulah wkwkwkðŸ˜,
user012	@marco_marconah makanan nya enk banget pokoknya mana nasi nya banyak lagiâˆ”i, •
user013	@dwiiiiintann mantap... katanya kalau lauk ayam, ayamnya separoh ya? Tp yg belum netas..
user014	@heni_fatmaa btw gk di sediakan kompor juga kah di Ma'had?
.....
user561	Bagaimana tata cara pembayaran pema'hadan ? @pmb.uinws @uinwalisongosemarang
user562	jalur snbp kok UKTnya setara kaya jalur mandiri wkwk
user563	Min mau tanya. Untuk daftar nama-nama mahasiswa/i yang masuk Ma'had di semester 1 atau 2 bisa dilihat dimana ya? Terimakasih
user564	Pening kepala
user565	Lo lo lo lo ga bahaya ta?

Lampiran 2. Contoh File Yang Sudah Diberi Label

Komentar	Label
mandek mondok pengen kuliah, eh malah dikon mondok meneh anjirrr	negatif
@imchoice19__ betuuulll	negatif
Akwoksowk welcome to ta'dziran tiap minggu	negatif
Itu segitu ada wajib bayar makan lagi apa ga?	positif
@heni_fatmaa belum.. ada bayar catering dan laundry	positif
@marco_marconah serius?	negatif
@marco_marconah sekitar berapa?	positif
@wahyunugro hosp catering nya 450 k/ bulan terus laundry nya per kg 7 k	positif
@wahyunugro hosp Ya gak tau wkwkwk heheheh bisa2 intinya sabar aku juga pemaksaan kok	negatif
@nittcollage gmn komentarnya soal makanannya?	positif
@marco_marconah ya gitulah wkwkwk	negatif
@marco_marconah makanan nya enk banget pokoknya mana nasi nya banyak lagi	positif
@dwiiiintann mantap... katanya kalau lauk ayam, ayamnya separoh ya? Tp yg belum netas..	negatif
.....
Bagaimana tata cara pembayaran pema'hadan ? @pmb.uinws @uinwalisongosemarang	positif
jalur snbp kok UKTnya setara kaya jalur mandiri wkwk	negatif
Min mau tanya. Untuk daftar nama-nama mahasiswa/i yang masuk Ma'had di semester 1 atau 2 bisa dilihat dimana ya? Terimakasih	positif
Pening kepala	negatif
Lo lo lo lo ga bahaya ta?	negatif

Lampiran 3. Contoh File “Kamus bahasa tidak baku.csv”

Tidak Baku	Baku
aamiin	amin
about	tentang
account	akun
actually	sebenarnya
adekk	adik
adeku	adikku
adick	adik
adiks	adik
adix	adik
afaan	apaan
ajaaaa	aja
akwoksowk	tertawa
alloh	allah
ama	sama
and	dan
angel	susah
astagfirullah	istigfar
astaghfirullah	istigfar
atasano	atasi
auto	otomatis
awokawok	tertawa
ayoo	ayo
bacot	banyak berbicara
.....
yaw	ya
yo	ya
yo	ya
you	kamu
yowis	ya sudah
yukk	ayo
yws	ya sudah
zonk	mengecewakan

Lampiran 4. Contoh File “kata tidak baku 2.csv”

slang	makna
aaa	apa
abis	habis
ad	ada
afah	apa
aga	agak
aiyam	ayam
aiyamniya	ayamnya
ajaa	saja
akuu	aku
amat	sekali
apaa	apa
apaaa	apa
apasi	apa
aq	saya
are	apakah
ati	hati
ati	hati
ayamniya	ayamnya
azzaa	saja
baee	saja
baiiyar	bayar
baikk	baik
baiyar	bayar
.....
pokoke	pokoknya
kakakk	kakak
diguyuu	ditertawakan
biodatadiri	biodata diri
pns	pegawai negeri sipil
kakanya	kakaknya
dapet	dapat
nerapin	menerapkan

Lampiran 5. Contoh File “Stopwords.txt”

lah sip biar gimana ora rasah gawe su www reco terustp aja ya
kayaknya loh wa konoha miliarantp kek gembel beginierror
pokonamah persib weh mi bangke nya aalowa mentok pas vgr
biofarmadan yah ko lola huhhhh mulupayah yo kjj jjjk
nnmenjadikannya kemana kesini sih is the cok buru gara nob tuk
an aja down munyer thokkk nomor very for us log in kebanyakan
pas kali muluhp ram bos gajelasdikit susahdikit asus pisan loba
iya kita ln gentinglama perbaikannyarespon bolak balik ktp
hahassyyuuuu udah unfaedahdaftar ui hmmzzz ehh mo hey
update halo udh no telp mending oon ti gw keja mah dll noktp
this ricekhadehbagaimana nimuter puterpusiang scannysuhin
malah sendiri terus hadeh dl kemanamana gakjelas diupdatemalah
kebabanyakan masuklog tibatiba out gajelas qr bbm karnan boss
.....
insyallah pondokkupemikirannyacumacuancuaksss dh adaa rt cc
departemenagamari depag depagri ri toefl imka memories jpg
ilahi untu rohmam wkwj wkwwk wkkkk toefle mozak mazik
mungguh punyat polll now broo bgtu lhoo anjai pema sedta
zaenalmuttaqin syafir toefl vs prtm pratama da al taufik raditya
sahrul a oalah wk lololo dinakt tertawahah u nungget s ns izzul
liii pan sbr cuaks m chan ra eaak coy tarak iki hahaha heh dwix
abss alfiiee jojoo chuaks af billa ng khoirum sii eaa skalaskasming
pokoknyamah lailyy jendil ar hehehe heheh purbaa yaaa p bah
noor arang iki mun hemm wow heheh hmmm yuk ekhem ba doi
wkwkwwkwk toganjelto gituu chuaksss m b dx susudah ayy
kpn uwin pgn wkwkwwkw turininuktuinws yaaa hahahaaaa
meme elf killing of husnay sayaaa bpk beber ealah merek
yhbebas win ada

Lampiran 6. Dokumen Hasil Klasifikasi NBC Dengan SMOTE

komentar	label	prediksi
asli tertawa pondok pondok suka ajar kitab sesuai kampus nambah ahli bicara lancar bahasa inggris tertawa dongeng anak pondok luar pakai	positif	positif
sabar paksa tahun semester enak masuk lulus pondok	positif	negatif
bocor timbang	positif	positif
tata bayar pondok universitas islam negeri walisongo semarang	positif	positif
komen racun akun privasi malu tertawa	negatif	negatif
laboratorium komputer habis pikir	negatif	negatif
ekonomi admin kartu indonesia pintar kuliah	positif	positif
makan enak pokok nasi	positif	positif
habis pikir nalar	negatif	negatif
adik buka jasa titip tenang	positif	positif
universitas islam negeri terap wajib pondok semester	positif	positif
orang	negatif	negatif
admin daftar nama nama mahasiswa masuk pondok semester mana terimakasih	positif	negatif
murah	positif	positif
cepat sembuh universitas islam negeri walisongo	positif	positif
atas kirim kopi trek mudah antuk rektor	negatif	positif
sistem universitas islam negeri walisongo layak survei temu uang kuliah tunggal sesuai pengaruh	negatif	negatif
serah pasu anak kerah	positif	positif
masuk guru negeri murah wajib pondok pakai uang	negatif	positif
main cantik	negatif	negatif

henti pondok kuliah suruh pondok	negatif	negatif
uang kuliah tunggal elit fasilitas sulit	negatif	negatif
sedia kompor pondok	positif	negatif
syarat uji skripsi	negatif	negatif
model main tertawa	negatif	positif
universitas islam negeri pimpin kacau peras finansial tinas intelektual	negatif	negatif
alumni pondok	positif	positif
syarat kartu rencana studi skripsi lucu	negatif	positif
tujuan dapat	positif	positif
berita pondok sebar kecewa bilang ubah	negatif	negatif
uang kuliah tunggal langit fasilitas sulit	negatif	negatif
aduh hati bocor kucing	negatif	positif
masuk pondok	negatif	positif
calon mahasiswa universitas islam negeri walisono lulus universitas islam negeri jember kondisi ayah wafat kerja ada jual ayam geprek rumah hasil bulan alhamdulillah kuliah dapat beasiswa	positif	negatif
takut tertawa	negatif	positif
adekku suruh kuliah masuk wajib bayar pondok semester mahal keluar susah swasta	negatif	negatif
universitas islam negeri wajib pondok	positif	positif
komisi berantas korupsi	negatif	negatif
untung teman teman	negatif	positif
doa sistem tertawa	positif	positif
ujung orang masuk lembaga didik jebak bisnis industri didik	negatif	negatif
innalillahi umum bayar uang kuliah tunggal jalur seleksi nasional dasar prestasi	negatif	negatif
tolong komen takut jebak universitas islam negeri walisono semarang	negatif	negatif
tertawa	negatif	positif
selamat makan hati	negatif	negatif

tertawa orang makan uang haram kecil doain ayo tuhan marah	negatif	positif
---	---------	---------

Lampiran 7. Dokumen Hasil Klasifikasi NBC Dengan ADASYN

komentar	label	prediksi
asli tertawa pondok pondok suka ajar kitab sesuai kampus nambah ahli bicara lancar bahasa inggris tertawa dongeng anak pondok luar pakai	positif	positif
sabar paksa tahun semester enak masuk lulus pondok	positif	negatif
bocor timbang	positif	positif
tata bayar pondok universitas islam negeri walisongo semarang	positif	positif
komen racun akun privasi malu tertawa	negatif	negatif
laboratorium komputer habis pikir	negatif	negatif
ekonomi admin kartu indonesia pintar kuliah	positif	positif
makan enak pokok nasi	positif	positif
habis pikir nalar	negatif	negatif
adik buka jasa titip tenang	positif	positif
universitas islam negeri terap wajib pondok semester	positif	positif
orang	negatif	positif
admin daftar nama nama mahasiswa masuk pondok semester mana terimakasih	positif	negatif
murah	positif	positif
cepat sembuh universitas islam negeri walisongo	positif	positif
atas kirim kopi trek mudah antuk rektor	negatif	positif
sistem universitas islam negeri walisongo layak survei temu uang kuliah tunggal sesuai pengaruh	negatif	negatif
serah pasu anak kerah	positif	positif
masuk guru negeri murah wajib pondok pakai uang	negatif	positif
main cantik	negatif	negatif

henti pondok kuliah suruh pondok	negatif	negatif
uang kuliah tunggal elit fasilitas sulit	negatif	negatif
sedia kompor pondok	positif	positif
syarat uji skripsi	negatif	negatif
model main tertawa	negatif	negatif
universitas islam negeri pimpin kacau peras finansial tinas intelektual	negatif	negatif
alumni pondok	positif	positif
syarat kartu rencana studi skripsi lucu	negatif	positif
tujuan dapat	positif	negatif
berita pondok sebar kecewa bilang ubah	negatif	negatif
uang kuliah tunggal langit fasilitas sulit	negatif	negatif
aduh hati bocor kucing	negatif	positif
masuk pondok	negatif	positif
calon mahasiswa universitas islam negeri walisono lulus universitas islam negeri jember kondisi ayah wafat kerja ada jual ayam geprek rumah hasil bulan alhamdulillah kuliah dapat beasiswa	positif	negatif
takut tertawa	negatif	positif
adekku suruh kuliah masuk wajib bayar pondok semester mahal keluar susah swasta	negatif	negatif
universitas islam negeri wajib pondok	positif	positif
komisi berantas korupsi	negatif	negatif
untung teman teman	negatif	positif
doa sistem tertawa	positif	positif
ujung orang masuk lembaga didik jebak bisnis industri didik	negatif	negatif
innalillahi umum bayar uang kuliah tunggal jalur seleksi nasional dasar prestasi	negatif	positif
tolong komen takut jebak universitas islam negeri walisono semarang	negatif	negatif
tertawa	negatif	positif
selamat makan hati	negatif	negatif

tertawa orang makan uang haram kecil doain ayo tuhan marah	negatif	positif
---	---------	---------

Lampiran 8. Source Code

Import Library

```
import string
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import nltk

import re, emoji

import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

from warnings import filterwarnings
filterwarnings('ignore')

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

Insert Dataset

```
data = pd.read_csv('DataKomentar.csv', sep=',')
data.head()
```

Data Preparation

Case Folding

```
def casefolding(komentar):
    komentar = komentar.lower()
    return komentar
data['Lower'] = data['Komentar'].apply(casefolding)
data.head()
```

Cleansing

```
def cleansing(komentar):
    komentar = komentar.strip(" ")
    komentar = re.sub(r'[?!\.]+(?:=.\!])', '', komentar)
    komentar = re.sub('\B@w+', '', komentar)
    komentar = re.sub('[^a-zA-Z]', '', komentar)
    komentar = emoji.demojize(str(komentar))
    komentar = komentar.strip(" ")
    return komentar
data['Clean'] = data['Lower'].apply(cleansing)
data.head(10)
```

Normalization

```
kamus_slang = pd.read_csv('Kamus bahasa tidak baku.csv', sep=',')
slang = pd.Series(kamus_slang['Baku'].values, index = kamus_slang['Tidak Baku']).to_dict()
kamus_slang.head()
```

```
def Slangwords(komentar):
    for word in komentar.split():
        if word in slang.keys():
            komentar = komentar.replace(word, slang[word])
            komentar = re.sub('@[w]+', '', komentar)
    return komentar
data['Slang1'] = data['Clean'].apply(Slangwords)
data.to_csv('data_normal.csv')
data.head()
```

```
kamus_slang2 = pd.read_csv('kata tidak baku 2.csv')
slang2 = pd.Series(kamus_slang2['makna'].values, index = kamus_slang2['slang']).to_dict()
kamus_slang2.head()
```

```
def Slangwords(komentar):
    for word in komentar.split():
        if word in slang2.keys():
            komentar = komentar.replace(word, slang2[word])
            komentar = re.sub('[\w]+', '', komentar)
    return komentar
data['Slang2'] = data['Slang1'].apply(Slangwords)
data.to_csv('data_normal2.csv')
data.head()
```

Tokenization

```
def token(text):
    tokenisasi = text.split()
    return tokenisasi
data['Tokenize'] = data['Slang2'].apply(token)
data.head()
```

Stopword Removal

```
stop = stopwords.words('indonesian')
txt_stopword = pd.read_csv('stopwords.txt', names= ["stopwords"], header = None)
stop.extend(txt_stopword['stopwords'][0].split(' '))

def stopword(hasil_token):
    return [word for word in hasil_token if word not in stop]
data['Stopword'] = data['Tokenize'].apply(lambda x: stopword(x))
data.to_csv('data_stopword.csv')
data.head()
```

Stemming

```
def stemming(komentar):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    x = []
    for word in komentar:
        komen = stemmer.stem(word)
        x.append(komen)
    komen_bersih = []
    komen_bersih = ' '.join(x)
    return komen_bersih
data['Stemming'] = data['Stopword'].apply(stemming)
data.head()
```

Remove Comment

```
def remove_comments(komentar):
    zero_length_comments = komentar[data["Stemming"].map(len) == 0]
    zero_length_comments_index = [ind for ind in zero_length_comments.index]
    data.drop(zero_length_comments_index, inplace = True)
    return data

data = remove_comments(data)
data.to_csv('data_clean.csv')
data.head(10)
```

Split Validation

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data['Stemming'], data['Label'], test_size= 0.1, random_state= 42)
X_train_dataframe = pd.DataFrame({'komentar': X_train, 'label':y_train})
X_train_dataframe.to_csv('X_Train_Dataset.csv')

print('X_train dataset: ', X_train.shape)
print('y_train dataset: ', y_train.shape)
print('X_test dataset: ', X_test.shape)
print('y_test dataset: ', y_test.shape)
```

TF IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer

tf = TfidfVectorizer()
text_tf = tf.fit_transform(data['Stemming'])

X_train_tfidf = tf.transform(X_train)
X_test_tfidf = tf.transform(X_test)

X_train_df = pd.DataFrame(X_train_tfidf.toarray(), columns=tf.get_feature_names_out())
X_train_df.to_csv('X_train_TFIDF.csv')

X_train_tfidf.toarray()
```

Oversampling SMOTE

```
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state = 42)
X_train_smote, y_train_smote, = smote.fit_resample(X_train_tfidf, y_train)

smote_df = pd.DataFrame(X_train_smote.toarray(), columns=tf.get_feature_names_out(), index=y_train_smote)
smote_df.to_csv('Hasil_SMOTE.csv')

print("Sebelum oversampling, jumlah label 'positif': {}".format(sum(y_train == 'positif')))
print("Sebelum oversampling, jumlah label 'negatif': {}".format(sum(y_train == 'negatif')))

print("Setelah oversampling, bentuk train_X: {}".format(X_train_smote.shape))
print("Setelah oversampling, bentuk train_y: {}".format(y_train_smote.shape))

print("Setelah oversampling, jumlah label 'positif': {}".format(sum(y_train_smote == 'positif')))
print("Setelah oversampling, jumlah label 'negatif': {}".format(sum(y_train_smote == 'negatif')))
```

Modeling, Evaluasi

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

from jcopml.plot import plot_classification_report, plot_confusion_matrix

clf_sm = MultinomialNB().fit(X_train_smote, y_train_smote)
predicted_sm = clf_sm.predict(X_test_tfidf)

sentimen_smote_df = pd.DataFrame({'komentar': X_test, 'label': y_test, 'prediksi': predicted_sm})
sentimen_smote_df.to_csv('Sentimen_SMOTE.csv', index=False)

print("MultinomialNB Accuracy:", accuracy_score(y_test,predicted_sm))
print("MultinomialNB Precision:", precision_score(y_test,predicted_sm, average='weighted'))
print("MultinomialNB Recall:", recall_score(y_test, predicted_sm, average='weighted'))
print("MultinomialNB f1_score:", f1_score(y_test,predicted_sm, average='weighted'))

class_label_sm = ["negative", "positive"]
df_cm_sm = pd.DataFrame(confusion_matrix(y_test, predicted_sm), index=class_label_sm, columns=class_label_sm)
sns.heatmap(df_cm_sm, annot=True, fmt='d')
plt.title("Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()

print('-----\n')
print(classification_report(y_test, predicted_sm, zero_division=0))
print('-----\n')
```

Oversampling ADASYN

```
from imblearn.over_sampling import ADASYN
adasyn = ADASYN(random_state = 42)
X_train_adasyn, y_train_adasyn, = adasyn.fit_resample(X_train_tfidf, y_train)

adasyn_df = pd.DataFrame(X_train_adasyn.toarray(), columns=tf_get_feature_names_out(), index=y_train_adasyn)
adasyn_df.to_csv('Hasil_ADASYN.csv')

print("Sebelum oversampling, jumlah label 'positif': {}".format(sum(y_train == 'positif')))
print("Sebelum oversampling, jumlah label 'negatif': {}".format(sum(y_train == 'negatif')))

print("Setelah oversampling, bentuk train_X: {}".format(X_train_adasyn.shape))
print("Setelah oversampling, bentuk train_y: {}".format(y_train_adasyn.shape))

print("Setelah oversampling, jumlah label 'positif': {}".format(sum(y_train_adasyn == 'positif')))
print("Setelah oversampling, jumlah label 'negatif': {}".format(sum(y_train_adasyn == 'negatif')))
```

Modeling, Evaluasi

```
clf_ad = MultinomialNB().fit(X_train_adasyn, y_train_adasyn)
predicted_ad = clf_ad.predict(X_test_tfidf)

hasil_df = pd.DataFrame({'komentar': X_test, 'label': y_test, 'prediksi': predicted_ad})
hasil_df.to_csv('Sentimen_ADASYN.csv', index=False)

print("MultinomialNB Accuracy:", accuracy_score(y_test,predicted_ad))
print("MultinomialNB Precision:", precision_score(y_test,predicted_ad, average='weighted'))
print("MultinomialNB Recall:", recall_score(y_test, predicted_ad, average='weighted'))
print("MultinomialNB f1_score:", f1_score(y_test,predicted_ad, average='weighted'))

class_label_ad = ["negative", "positive"]
df_cm_ad = pd.DataFrame(confusion_matrix(y_test, predicted_ad), index=class_label_ad, columns=class_label_ad)
sns.heatmap(df_cm_ad, annot=True, fmt='d')
plt.title("Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()

print('-----\n')
print(classification_report(y_test, predicted_ad, zero_division=0))
print('-----\n')
```

Visualization

```
DataWordCloud = pd.read_csv('data_clean.csv', sep=',')

sizes = [count for count in DataWordCloud['Label'].value_counts()]
labels = list(DataWordCloud['Label'].value_counts().index)
colors = ['lightblue', 'lightgreen']
plt.pie(sizes, labels = labels, colors = colors, autopct = '%1.1f%%', startangle = 90)
plt.title("Sentimen Pada Instagram \n Program Pemahadan 2023 \n (565)")
plt.show()
```

```
from wordcloud import WordCloud, STOPWORDS

comment_words = ''
stopwords = set(STOPWORDS)

for val in DataWordCloud['Stemming']:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "

wordcloud = WordCloud(width = 800, height = 800,
                      background_color = 'white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)

plt.figure(figsize = (10, 10), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

```

df_negative = DataWordCloud[ (DataWordCloud['Label']==' negatif')]
df_positive = DataWordCloud[ (DataWordCloud['Label']==' positif')]

#convert to list
negative_list = df_negative['Stemming'].tolist()
positive_list = df_positive['Stemming'].tolist()

filtered_negative = ("").join(str(negative_list))
filtered_negative = filtered_negative.lower()

filtered_positive = ("").join(str(positive_list))
filtered_positive = filtered_positive.lower()

```

WordCloud Positive

```

wordcloud = WordCloud(max_font_size= 160, margin= 0, background_color= "black", colormap= "Blues").generate(filtered_positive)
plt.figure(figsize= [10,10])
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.margins(x=0, y=0)
plt.title("Positive Comment WordCloud")
plt.show()

```

WordCloud Negative

```

wordcloud = WordCloud(max_font_size= 160, margin= 0, background_color= "black", colormap= "Reds").generate(filtered_negative)
plt.figure(figsize= [10,10])
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.margins(x=0, y=0)
plt.title("Negative Comment WordCloud")
plt.show()

```


Lampiran 10. Contoh Data Hasil *Oversampling* ADASYN

J	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Label	ada																			
2	postif	0	0	0	0	0	0	0.256248	0	0	0	0	0	0	0	0	0	0	0	0	0
3	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	negatif	0	0	0.849329	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	negatif	0	0	0	0	0	0	0.354772	0	0	0	0	0	0.369842	0	0	0	0	0	0	0
11	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	negatif	0	0	0	0	0	0	0	0	0.230774	0	0	0	0	0	0	0	0	0	0	0
13	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	postif	0	0	0	0	0.251566	0.264307	0	0	0	0	0	0	0.278664	0	0	0	0	0	0	0
25	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

J	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
406	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
407	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
408	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
409	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
410	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
411	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
412	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
413	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
414	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
415	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
416	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
417	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
418	negatif	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
419	negatif	0	0	0	0	0	0	0	0	0	0.724114	0	0	0	0	0	0	0	0	0	0
420	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
421	negatif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
422	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
423	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
424	postif	0	0	0	0	0.1771	0.186	0	0	0	0	0	0.6193271	0	0	0	0	0	0	0	0
425	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
426	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
427	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
428	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
429	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
430	postif	0	0	0	0	0.217165	0	0	0	0	0	0	0	0.238995	0	0	0	0	0	0	0

J	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
475	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
476	postif	0	0	0	0	0.300019	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
477	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
478	postif	0	0	0	0	0	0	0	0.05495	0	0	0	0	0	0	0	0	0	0	0	0
479	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
480	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
481	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
482	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
483	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
484	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
485	postif	0	0	0	0	0	0.138849	0	0.154897	0	0	0	0	0	0	0	0	0	0	0.180124	0
486	postif	0	0	0.73782	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
487	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
488	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
489	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
490	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
491	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
492	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
493	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
494	postif	0	0	0	0	0	0	0	0.251193	0	0	0	0	0	0	0	0	0	0	0	0
495	postif	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
496	postif	0	0</																		

RIWAYAT HIDUP

A. Identitas Diri

1. Nama Lengkap : Viorella Amanda Putri
2. Tempat dan Tanggal Lahir : Ngawi, 10 September 2002
3. Alamat Rumah : RT03/RW 02, Dusun Bendo, Desa Kauman, Kec. Sine, Kab. Ngawi, Jawa Timur
4. Nomor HP : 081554361851
5. E-mail : viorellaamandaputri10@gmail.com

B. Riwayat Pendidikan

1. Pendidikan Formal :
- TK Dharma Wanita Kauman
 - SDN Kauman
 - MAN 4 Ngawi
 - UIN Walisongo Semarang
2. Pendidikan Nonformal :
- -

C. Prestasi Akademik

-
-

D. Karya Ilmiah

-
-

Semarang, 1 April 2024



Viorella Amanda Putri
NIM 2008096025