

SKRIPSI
ANALISIS SENTIMEN TWITTER TERHADAP PERNIKAHAN DI USIA
MUDA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE
(SVM)



RADEN ISNAWAN ARGY ARYASATYA

NIM: 195410257

PROGRAM STUDI INFORMATIKA
PROGRAM SARJANA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA
YOGYAKARTA

2023

SKRIPSI

**ANALISIS SENTIMEN TWITTER TERHADAP PERNIKAHAN DI USIA
MUDA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE
(SVM)**

Diajukan sebagai salah satu syarat untuk menyelesaikan studi



Program Sarjana

Program Studi Informatika

Fakultas Teknologi Informasi

Universitas Teknologi Digital Indonesia

Yogyakarta

Disusun Oleh

RADEN ISNAWAN ARGY ARYASATYA

NIM: 195410257

PROGRAM STUDI INFORMATIKA

PROGRAM SARJANA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

YOGYAKARTA

2023

HALAMAN PERSETUJUAN

**Judul : Analisis Sentimen Twitter terhadap Pernikahan di
Usia Muda menggunakan Metode Support Vector
Machine (SVM)**

Nama : Raden Isnawan Argi Aryasatya

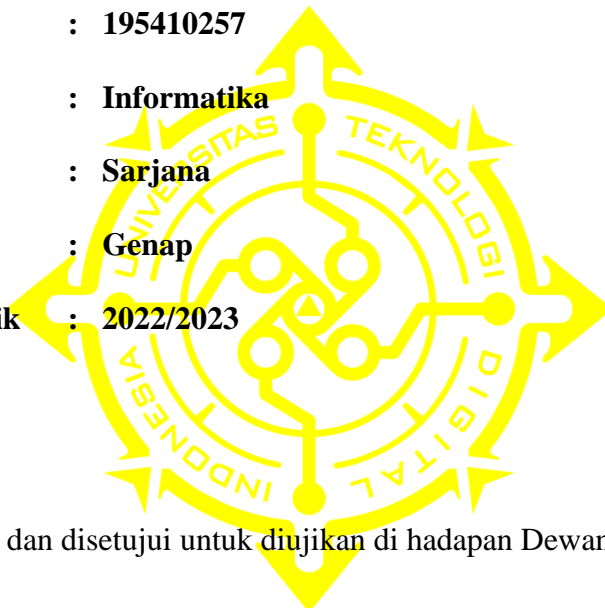
NIM : 195410257

Program Studi : Informatika

Program : Sarjana

Semester : Genap

Tahun Akademik : 2022/2023



Telah diperiksa dan disetujui untuk diujikan di hadapan Dewan Penguji Skripsi

Yogyakarta, 10 Juli 2023

Dosen Pembimbing,

Maria Mediatrix Sebatubun, S.Kom., M.Eng.

NIDN: 0514089101

HALAMAN PENGESAHAN

SKRIPSI

ANALISIS SENTIMEN TWITTER TERHADAP PERNIKAHAN DI USIA MUDA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM)

Telah dipertahankan di depan Dewan Penguji Skripsi dan dinyatakan
diterima untuk memenuhi sebagian persyaratan guna memperoleh Gelar

Sarjana Komputer
Program Studi Informatika
Fakultas Teknologi Informasi
Universitas Teknologi Digital Indonesia
Yogyakarta

Yogyakarta, 21 Juli 2023

Dewan Penguji	NIDN	Tanda Tangan
1. Sari Iswanti, S.Si., M. Kom.	0508027202
2. Ariesta Damayanti, S.Kom., M.Cs.	0020047801
3. Maria Mediatrix Sebatubun, S.Kom., M.Eng.	0514089101

Mengetahui
Ketua Program Studi Informatika

Dini Fakta Sari, S.T., M.T.
NPP : 121172

PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa naskah skripsi ini belum pernah diajukan untuk memperoleh gelar Sarjana Komputer di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara sah diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 3 Agustus 2023



Raden Isnawan Argi Aryasatya

NIM: 195410257

HALAMAN PERSEMBAHAN

Alhamdulillah rabbil'alamin. Puji dan syukur penulis ucapkan kepada Allah SWT yang telah memberikan berkah dan karunia-Nya sehingga penulis dapat dengan lancar mengerjakan skripsi ini. Karya tulis ini dipersembahkan kepada:

1. Bambang Hanggoro Heru Nurcahyo dan Wernita Tampubolon selaku orang tua penulis yang selalu memberikan kasih sayang dan dukungan kepada penulis di segala kondisi.
2. Fidella Azra Daniswara selaku adik kandung penulis yang telah mengajari banyak hal tentang tren di media sosial dan pengetahuan umum lainnya kepada penulis.
3. Dr. Bambang Purnomosidi P. D. P., Akt., S.Kom. MMSI. selaku dosen sekaligus paman penulis yang merupakan seorang *role model*, contoh, dan teladan bagi penulis dalam proses menggapai cita-cita.
4. Seluruh keluarga besar saya yang selalu memberikan dukungan dan semangat kepada saya untuk meraih kesuksesan.
5. Almarhum kakek dan almarhumah nenek saya yang sekarang sudah tenang di surga.
6. Sahabat – sahabat saya dari SMP Negeri 1 Yogyakarta lulusan tahun 2015 dan SMA Negeri 5 Yogyakarta lulusan tahun 2018 yang selalu hadir saat saya mengajak mereka untuk berkumpul dan saling tertawa bersama.

7. Sahabat – sahabat saya dari Politeknik Keuangan Negara STAN angkatan 2018 yang telah memberikan kenangan - kenangan paling indah dalam hidup saya.
8. Teman – teman dari Universitas Teknologi Digital Indonesia, terutama teman – teman dari kelas IF-3 dan IF-5 yang telah menerima saya ke dalam lingkungan pertemanan mereka dan membuat saya merasa merasa tidak kesepian saat pertama kali menginjakkan kaki di kampus tercinta.

HALAMAN MOTTO

“Life shrinks or expands in proportion to one's courage.”

- Anais Nin -

“To persevere is important for everybody. Don't give up, don't give in. There is always an answer to everything.”

- Louis Zamperini -

“With great power, comes great responsibility.”

- Uncle Ben (Spider-Man, 2002) -

KATA PENGANTAR

Senantiasa penulis ucapkan puji dan syukur atas rahmat dan kehadiran Allah SWT yang telah memberikan rahmat dan karunia-Nya, sehingga skripsi berjudul “Analisis Sentimen Twitter terhadap Pernikahan di Usia Muda menggunakan Metode Support Vector Machine (SVM)” dapat disusun dan diselesaikan dengan baik.

Skripsi ini disusun sebagai salah satu syarat untuk mendapatkan gelar strata-1 (S1) di program studi Informatika, Fakultas Teknologi Informasi, Universitas Teknologi Digital Indonesia. Dalam pengerjaan skripsi ini, penulis mendapatkan banyak dukungan dan bantuan dari beberapa pihak. Dalam kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Bapak Ir. Totok Suprawoto, M.M., M.T., selaku Rektor Universitas Teknologi Digital Indonesia.
2. Bapak Ir. Muhammad Guntara, M.T., selaku Dekan Fakultas Teknologi Informasi.
3. Ibu Dini Fakta Sari, S.T, M.T., selaku Ketua Program Studi Informatika Universitas Teknologi Digital Indonesia.
4. Ibu Maria Mediatrix Sebatubun, S.Kom., M.Eng., selaku dosen pembimbing yang telah meluangkan waktunya untuk memberikan bimbingan selama berlangsungnya kegiatan penyusunan skripsi.
5. Kedua orang tua tercinta dan seluruh keluarga yang selalu mendukung saya.

6. Ibu Ariesta Damayanti, S.Kom., M.Cs. dan Ibu Sari Iswanti, S.Si., M.Kom., selaku dosen penguji yang telah memberikan arahan dan koreksi bagi penulis dalam menyusun skripsi.
7. Bapak Drs. Tri Prabawa, M.Kom., selaku dosen pembimbing PKL yang telah memberi ilmu kepada penulis tentang cara membuat karya tulis yaitu laporan PKL sehingga penulis dapat menyusun skripsi dengan lancar.
8. Seluruh dosen di Universitas Teknologi Digital Indonesia yang telah memberikan ilmunya kepada penulis selama perkuliahan di kampus ini.
9. Seluruh civitas akademika Universitas Teknologi Digital Indonesia yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi dan menyelesaikan skripsi di Universitas Teknologi Digital Indonesia.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini. Oleh karena itu, penulis mengucapkan mohon maaf atas kesalahan yang terdapat di dalam skripsi ini. Penulis juga mengharapkan pembaca untuk menyampaikan kritik dan saran kepada penulis skripsi ini sebagai sarana introspeksi diri bagi penulis dalam membuat karya tulis di masa yang akan datang. Penulis berharap laporan ini dapat bermanfaat bagi para pembaca. Terima kasih.

Yogyakarta, 3 Agustus 2023



Raden Isnawan Argi Aryasatya

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
INTISARI.....	xvii
ABSTRACT.....	xviii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Ruang Lingkup	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	5

BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI	7
2.1 Tinjauan Pustaka	7
2.2 Dasar Teori	11
2.2.1 Twitter	11
2.2.2 Snsrape	12
2.2.3 Python	12
2.2.4 Data Mining	13
2.2.5 Machine Learning	14
2.2.6 Analisis Sentimen	16
2.2.7 Text Preprocessing	16
2.2.8 Pelabelan Data.....	18
2.2.9 Ekstraksi Fitur	19
2.2.10 Support Vector Machine	21
2.2.11 Evaluasi Performansi	24
BAB 3 METODE PENELITIAN.....	26
3.1 Bahan/Data	26
3.1.1 Kebutuhan Input.....	26
3.1.2 Kebutuhan Proses.....	26
3.1.3 Kebutuhan Output	27
3.2 Peralatan	27

3.2.1	Kebutuhan Perangkat Lunak	27
3.2.2	Kebutuhan Perangkat Keras	27
3.3	Prosedur Pengumpulan Data	28
3.4	Analisis dan Rancangan Sistem.....	28
3.4.1	Block Diagram	28
3.4.2	Flowchart Support Vector Machine	31
3.4.3	Perancangan Antarmuka	34
BAB 4 IMPLEMENTASI DAN PEMBAHASAN SISTEM		38
4.1	Implementasi	38
4.1.1	Pengambilan Data	38
4.1.2	Preprocessing	39
4.1.3	Labeling.....	45
4.1.4	Ekstraksi Fitur	48
4.1.5	Implementasi SVM	49
4.1.6	Evaluasi Performansi	50
4.2	Uji Coba dan Pembahasan Sistem.....	54
4.2.1	Antarmuka Aplikasi Web.....	54
4.2.2	Hasil Uji Coba.....	57
BAB 5 PENUTUP		60
5.1	Kesimpulan.....	60

5.2	Saran	60
DAFTAR PUSTAKA		62
LAMPIRAN		

DAFTAR GAMBAR

Gambar 2.1 Tahapan <i>Knowledge Discovery Database</i> (Ependi & Putra, 2019) ..	14
Gambar 2.2 Skema <i>Artificial Intelligence</i> dan <i>Machine Learning</i> (Roihan, et al., 2020)	15
Gambar 2.3 <i>Text Preprocessing</i>	17
Gambar 2.4 Support Vector Machine (Parapat, et al., 2018)	21
Gambar 2.5 Pemetaan <i>Input Space</i> Berdimensi Dua dengan Pemetaan ke Dimensi Tinggi (Rahutomo et al., 2018)	23
Gambar 3.1 Block Diagram	28
Gambar 3.2 Flowchart SVM	31
Gambar 3.3 Visualisasi <i>Hyperplane</i>	34
Gambar 3.4 Desain Halaman Beranda	35
Gambar 3.5 Desain Halaman Confusion Matrix	36
Gambar 3.6 Desain Halaman Classification Report	36
Gambar 3.7 Desain Halaman Sebaran Data	37
Gambar 4.1 Kode <i>Scraping</i> Bulan Maret	38
Gambar 4.2 Kode <i>Scraping</i> Bulan April	39
Gambar 4.3 Kode <i>Case Folding</i>	39
Gambar 4.4 Kode <i>Data Cleansing</i>	40
Gambar 4.5 Kode <i>Tokenizing</i>	41
Gambar 4.6 File Normalisasi	42
Gambar 4.7 Kode Normalisasi	42
Gambar 4.8 Kode <i>Filtering</i>	43

Gambar 4.9 Kode <i>Stemming</i>	44
Gambar 4.10 Kode Penggabungan Kata	45
Gambar 4.11 Google Spreadsheet.....	46
Gambar 4.12 Hasil Pelabelan dengan TextBlob	46
Gambar 4.13 <i>Bar Chart Labeling</i>	47
Gambar 4.14 <i>Pie Chart Labeling</i>	47
Gambar 4.15 Kode <i>Split</i> Data Latih dan Data Uji	48
Gambar 4.16 Kode <i>CountVectorizer</i>	48
Gambar 4.17 Kode Pelatihan dan Pengujian SVM.....	49
Gambar 4.18 Kode Akurasi SVM.....	49
Gambar 4.19 Kode <i>Confusion Matrix</i>	50
Gambar 4.20 Visualisasi <i>Confusion Matrix</i>	51
Gambar 4.21 Kode <i>Classification Report</i>	52
Gambar 4.22 <i>Classification Report</i>	53
Gambar 4.23 Halaman Beranda	55
Gambar 4.24 Halaman Confusion Matrix	56
Gambar 4.25 Halaman Classification Report.....	56
Gambar 4.26 Halaman Sebaran Data	57
Gambar 4.27 Input Teks dengan Hasil Klasifikasi Negatif	58
Gambar 4.28 Input Teks dengan Hasil Klasifikasi Netral	58
Gambar 4.29 Input Teks dengan Hasil Klasifikasi Positif.....	59

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	9
Tabel 2.2 Contoh DTM	20
Tabel 2.3 Contoh Penerapan <i>n-gram</i>	20
Tabel 2.4 <i>Confusion Matrix</i>	24
Tabel 3.1 Tabel Contoh Masukan Data Klasifikasi SVM	32
Tabel 3.2 Tabel Hasil Analisis Klasifikasi SVM	34
Tabel 4.1 Tabel <i>Case Folding</i>	39
Tabel 4.2 Tabel <i>Data Cleansing</i>	40
Tabel 4.3 Tabel <i>Tokenizing</i>	41
Tabel 4.4 Tabel Normalisasi	42
Tabel 4.5 Tabel <i>Filtering</i>	44
Tabel 4.6 Tabel <i>Stemming</i>	45
Tabel 4.7 Tabel Penggabungan Kata	45
Tabel 4.8 Tabel Hasil Pelabelan dengan TextBlob	47
Tabel 4.9 Tabel Perbandingan Akurasi Tiga Kernel SVM	50
Tabel 4.10 Tabel <i>Confusion Matrix</i>	51

INTISARI

Fenomena pernikahan di usia muda banyak terjadi di Indonesia. Pada tahun 2021, Komnas Perempuan mencatat ada 59.709 kasus pernikahan dini yang mendapat dispensasi. Hal itu membuktikan bahwa masih banyak masyarakat Indonesia yang tidak mengetahui atau tidak mengikuti aturan UU No. 1 tahun 1974 yang mengatur tentang umur minimal menikah yaitu 19 tahun. Topik ini pun selalu menjadi perbincangan hangat di Twitter. Ada netizen yang beropini baik, ada yang beropini jelek, dan ada pula yang netral. Hal tersebut dapat dianalisis menggunakan analisis sentimen.

Analisis sentimen dapat digunakan untuk menganalisis pendapat netizen dan mengelompokkannya menjadi tiga kategori yaitu positif, netral, dan negatif. Penelitian ini menggunakan metode *Support Vector Machine* (SVM) untuk melakukan klasifikasi yang dilakukan dengan garis pembatas (*hyperplane*) yang memisahkan kelas-kelas yang ada pada kumpulan data netizen Twitter yang memiliki opini tentang nikah muda.

Jumlah data yang digunakan sebanyak 4000 data yang diambil selama bulan Maret dan April 2023. Data dibagi dengan rasio 80:20 dimana 3200 data digunakan untuk data latih dan 800 data digunakan sebagai data uji. Penelitian ini menghasilkan akurasi *Linear Support Vector Machine* sebesar 87,375% dengan nilai presisi 82% untuk sentimen negatif, 40% untuk sentimen netral, dan 91% untuk sentimen positif. Selain itu juga dihasilkan nilai *recall* 64% untuk sentimen negatif, 44% untuk sentimen netral, dan 95% untuk sentimen positif. Terakhir, *f1-score* untuk sentimen negatif adalah 72%, lalu 42% untuk sentimen netral, kemudian 93% untuk sentimen positif.

Kata kunci: *analisis sentimen, nikah muda, support vector machine, twitter*

ABSTRACT

The phenomenon of marriage at a young age often occurs in Indonesia. In 2021, Komnas Perempuan stated that there were 59,709 cases of early marriage that received dispensations. This proves that there are still many Indonesian people who do not know or do not follow the UU No. 1 tahun 1974 which regulates the minimum age for marriage, which is 19 years old. This topic has always been a hot conversations topic on Twitter. There are netizens who have good opinions, there are those who have bad opinions, and there are those who are neutral. This can be analyzed using sentiment analysis.

Sentiment analysis can be used to analyze the opinions of netizens and group them into three categories, which are positive, neutral and negative. This study uses the Support Vector Machine (SVM) method to do the classification which is done with a dividing line (hyperplane) that separates the classes in the data set of Twitter netizens who have opinions about young marriage.

The data used in this research are 4000 data retrieved in March and April 2023. The data is divided by a ratio of 80:20, where 3200 data are used for training data and 800 data are used as testing data. This research produces a Linear Support Vector Machine accuracy of 87,375% with a precision value of 82% for negative sentiment, 40% for neutral sentiment, and 91% for positive sentiment. In addition, a recall value of 64% for negative sentiment, 44% for neutral sentiment, and 95% for positive sentiment were also generated. Finally, the f1-score for negative sentiment is 72%, then 42% for neutral sentiment, then 93% for positive sentiment.

Keywords: *sentiment analysis, support vector machine, twitter, young marriage*

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Salah satu fenomena sosial yang sering terjadi di Indonesia adalah masalah pernikahan di usia muda. Fenomena ini banyak terjadi di berbagai wilayah di tanah air, baik di perkotaan maupun di pedesaan. Menurut data milik Komnas Perempuan, sepanjang tahun 2021 ada 59.709 kasus pernikahan dini yang diberikan dispensasi oleh pengadilan (Harruma, 2022). Hal ini menunjukkan betapa sederhana pemikiran rakyat Indonesia terhadap pernikahan. Padahal, segala hal tentang pernikahan sudah ada aturan tertulisnya.

Peraturan tentang umur minimal untuk melakukan perkawinan diatur dalam pasal 7 ayat (1) Undang – Undang Republik Indonesia Nomor 16 Tahun 2019 tentang perubahan atas Undang – Undang Nomor 1 tahun 1974 tentang Perkawinan, yang berbunyi “Perkawinan hanya diizinkan apabila pria dan wanita sudah mencapai umur 19 (sembilan belas) tahun”. Jika masyarakat Indonesia menerapkan aturan tersebut dengan baik, maka tentunya dampak – dampak dari pernikahan di usia muda dapat diminimalisir. Dampak – dampak tersebut antara lain adalah anak *stunting*, anak putus sekolah, kekerasan dalam rumah tangga, perceraian, kematian bayi dan ibu saat melahirkan, anak menjadi beban orang tua karena belum mampu secara ekonomi, dan lain – lain (Serliana, 2020).

Berdasarkan dampak – dampak tersebut, maka muncul perdebatan terhadap kasus pernikahan di usia muda salah satunya di media sosial seperti Instagram, Facebook, dan terutama di Twitter. Di Twitter, banyak warganet yang memberikan

opini mereka terhadap pelaksanaan pernikahan di usia muda. Ada sebagian netizen yang pro, ada yang kontra, dan ada juga yang tidak memihak. Karena munculnya berbagai persepsi atau opini masyarakat tersebut, maka di penelitian ini dilakukan analisis sentimen yang diharapkan dapat dengan efektif mengumpulkan data-data berupa opini masyarakat tentang pernikahan usia muda.

Analisis sentimen atau bisa disebut juga dengan *opinion mining* adalah bidang studi yang merupakan cabang dari data mining yang berguna untuk menganalisis pendapat orang terhadap suatu entitas seperti layanan, organisasi, produk, individu, masalah, peristiwa, dan lain – lain. Analisis ini memiliki fokus pada pendapat yang mengekspresikan sentimen positif, netral, dan negatif (Septian, et al., 2018). Analisis sentimen dapat diterapkan menggunakan beberapa algoritma atau metode *machine learning*.

Dalam melakukan penelitian ini, digunakan metode *Support Vector Machine* (SVM) untuk melakukan klasifikasi yang dilakukan dengan garis pembatas (*hyperplane*) yang memisahkan tiga kelas yaitu kelas opini positif, opini netral, dan opini negatif (Haranto & Sari, 2019). Dalam penelitian ini, SVM dipilih karena memiliki beberapa kelebihan yang cocok dengan penelitian ini. Salah satu kelebihan metode SVM adalah bisa menghasilkan model klasifikasi yang baik walaupun hanya dilatih dengan data yang sedikit (Suyanto, 2017).

Selain itu, ada alasan lain juga yang menyatakan secara implisit bahwa SVM merupakan metode yang cukup fleksibel dan akurat untuk penelitian ini. Fleksibel karena proses klasifikasi dengan SVM dapat dilakukan dengan memilih salah satu di antara 4 kernel populer yang tersedia sesuai dengan himpunan data yang dimiliki

yaitu *linear*, *polynomial*, RBF, dan *sigmoid* dan dapat memanfaatkan *kernel trick* sampai mendapatkan *hyperplane* yang optimal (Husada & Paramita, 2021). Akurat karena klasifikasi sentimen dengan metode SVM memiliki akurasi yang cenderung lebih baik dibandingkan dengan metode – metode lain yang dapat dilihat pada beberapa referensi pada tinjauan pustaka penelitian ini yaitu penelitian yang berjudul “Perbandingan Akurasi Analisis Sentimen Tweet terhadap Pemerintah Provinsi DKI Jakarta di Masa Pandemi” (Himawan & Eliyani, 2021) yang mendapatkan akurasi SVM sebesar 77,58%, dan penelitian berjudul “Analisis Sentimen Perusahaan Listrik Negara Cabang Ambon Menggunakan Metode Support Vector Machine dan Naive Bayes Classifier” (Tuhuteru dan Iriani, 2018) yang mendapatkan akurasi SVM sebesar 81.67%.

Berdasarkan penelitian – penelitian sebelumnya yang sudah dipaparkan di tinjauan pustaka penelitian ini, maka akan dilakukan analisis sentimen pada *tweets* pengguna di media sosial Twitter tentang pernikahan di usia muda menggunakan *Support Vector Machine* (SVM) untuk menentukan klasifikasi antara sentimen positif dan negatif pada *tweets* tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sebelumnya sudah dipaparkan, maka rumusan masalah yang dapat diangkat pada penelitian ini adalah bagaimana cara melakukan analisis sentimen terhadap opini masyarakat mengenai pernikahan di usia muda menggunakan metode *Support Vector Machine*.

1.3 Ruang Lingkup

Ruang lingkup pada penelitian ini adalah:

1. Penelitian ini menggunakan media sosial Twitter sebagai sumber data.
2. Sumber data yang didapatkan dari Twitter merupakan tweet berbahasa Indonesia
3. Menggunakan Snsrape untuk mengambil *tweet* dari Twitter.
4. Data yang diambil berjumlah 4000 *tweet* yang diambil pada bulan Maret dan April 2023.
5. Analisis sentimen diterapkan menggunakan metode *Support Vector Machine* dengan kernel *Linear*.
6. Proses analisis data dilakukan menggunakan bahasa pemrograman Python dengan tool Jupyter Notebook dan Visual Studio Code.
7. Hasil analisis merupakan klasifikasi sentimen dengan kelas positif, netral, dan negatif.

1.4 Tujuan Penelitian

Tujuan dari dilakukannya penelitian ini adalah untuk mengklasifikasikan nilai sentimen dengan menentukan kelas positif, netral, dan negatif data *tweets* dari pengguna media sosial Twitter yang berkaitan dengan topik pernikahan di usia muda menggunakan algoritma *Support Vector Machine* (SVM), serta untuk mengetahui tingkat akurasi dan performa metode SVM dalam melakukan analisis sentimen.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Mengetahui pandangan masyarakat terhadap pernikahan di usia muda yang terjadi di Indonesia.

2. Hasil dari penelitian ini bisa memberikan presepsi dan dorongan kepada pemerintah untuk terus memberikan sosialisasi, penyuluhan, ataupun beberapa kebijakan kepada masyarakat supaya masyarakat bisa mengikuti aturan terkait umur minimal pernikahan.
3. Untuk mengetahui berapa banyak masyarakat yang pro dan yang kontra terhadap fenomena pernikahan usia muda.

1.6 Sistematika Penulisan

BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang, rumusan masalah, ruang lingkup, tujuan penelitian, manfaat penelitian, dan sistematika penelitian.

BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini membahas tentang sumber – sumber pustaka yang dijadikan acuan penelitian dan dasar teori yang menjadi dasar dalam penulisan skripsi.

BAB 3 METODE PENELITIAN

Bab ini membahas tentang setiap langkah penelitian yang meliputi bahan, peralatan, prosedur pengumpulan data, dan analisis rancangan sistem yang dijelaskan secara deskriptif menggunakan pemodelan diagram.

BAB 4 IMPLEMENTASI DAN PEMBAHASAN SISTEM

Bab ini menguraikan tentang implementasi sistem yang merupakan inti dari penelitian yang sesuai dengan rancangan berdasarkan komponen, *tools*, dan bahasa pemrograman yang sudah dituliskan pada bab sebelumnya.

BAB 5 PENUTUP

Bab ini berisi kesimpulan terhadap hasil penelitian yang dilakukan. Selain itu, pada bab ini juga diberikan saran untuk pengembangan sistem lebih lanjut.

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Ada beberapa penelitian yang membahas tentang analisis sentimen dengan berbagai metode yang digunakan sebagai acuan pembuatan penelitian ini. Di antaranya adalah penelitian yang dilakukan oleh Himawan & Eliyani (2021) yang membahas tentang perbandingan tiga algoritma *machine learning* yaitu algoritma *Random Forest Classifier*, *Naïve Bayes*, dan *Support Vector Machine* untuk melakukan penelitian berupa analisis sentimen terhadap Pemerintah Provinsi DKI Jakarta di Masa Pandemi dengan variabel nilai negatif, netral, dan positif. Penelitian ini menghasilkan hasil akurasi algoritma *Random Forest Classifier* sebesar 75,81%, algoritma *Naïve Bayes* dengan hasil akurasi 75,22%, dan algoritma *Support Vector Machine* 77,58%.

Selanjutnya adalah penelitian yang dilakukan oleh Simorangkir dan Lhaksamana (2018) mengenai analisis sentimen untuk Mobile Legends dan Arena of Valor di media sosial Twitter. Dengan menggunakan metode *Naïve Bayes Classifier*, peneliti menentukan klasifikasi *tweets* yang memiliki sentimen negatif dan positif. Hasilnya Mobile Legends memiliki 33 *tweet* positif dan 44 *tweet* negatif. Hasil akurasi, *error*, *recall* dan *precision* yang didapat untuk Mobile Legends masing-masing sebesar 88,89%, 19,18%, 96,97%, dan 69,57%. Sementara Arena of Valor memiliki 54 *tweet* positif dan 151 *tweet* negatif. Hasil nilai akurasi, *error*, *recall* dan *precision* untuk Arena of Valor masing-masing sebesar 39,02%, 60,98%, 88,89% dan 28,74%.

Penelitian berikutnya adalah penelitian yang dilakukan oleh Septian, Fahrudin, dan Nugroho (2019) mengenai analisis sentimen pada Twitter tentang polemik persepakbolaan di Indonesia. Penelitian ini menggunakan pembobotan TF-IDF dan klasifikasi dengan metode *K-Nearest Neighbor* pada 2000 data *tweet* berbahasa Indonesia yang memiliki kata kunci “@pssi”. Dari seluruh data tersebut, didapatkan hasil akurasi optimal pada nilai $k=23$ sejumlah 79,99%.

Pravina, Cholissodin, Adikara (2019) melakukan penelitian berupa analisis sentimen pada opini masyarakat di Twitter tentang maskapai penerbangan menggunakan metode *Support Vector Machine*. Sentimen analisis ini menerapkan fitur *Lexicon Based* untuk menerima opini berbahasa lain selain Bahasa Indonesia. Dengan menggunakan parameter C bernilai 10 dan learning rate bernilai 0,03 dan menggunakan *Lexicon Based Features* dengan iterasi sebanyak 50 kali, dapat dihasilkan *accuracy* sebesar 40%, *precision* 40%, 100% *recall*, dan *f-measure* sebesar 57,14%.

Kemudian Haranto dan Sari (2019) juga melakukan penelitian tentang implementasi *Support Vector Machine* untuk analisis sentimen tentang opini masyarakat terhadap pelayanan Telkom dan Biznet pada medias sosial Twitter. Penelitian ini menggunakan dataset sebanyak 500 *tweet* yang berasal dari *crawling* data Twitter, dan terdapat 250 *tweet* yang dijadikan *dataset* pada masing-masing objek. Penelitian ini menghasilkan nilai *accuracy* 79,6%, *precision* 76,5%, *recall* 72,8% , dan *F1-score* 74,6% untuk Telkom, serta *accuracy* 83,2%, *precision* 78,8%, *recall* 71,6%, dan *F1-score* 75% untuk Biznet.

Penelitian terakhir adalah penelitian yang dilakukan oleh Tuhuteru dan Iriani (2018) membahas tentang analisis sentimen mengenai kinerja PLN cabang Ambon menggunakan dua metode *machine learning* yaitu *Support Vector Machine* dan *Naïve Bayes Classifier*. Hasil perbandingan metode klasifikasi analisis sentimen pada kasus ini menunjukkan metode SVM lebih baik daripada NBC, dengan tingkat akurasi sebesar 81.67%. Sedangkan metode klasifikasi NBC hanya memiliki nilai akurasi sebesar 67.20%.

Tabel 2.1 Tinjauan Pustaka

No	Nama Peneliti	Judul Penelitian	Metode	Hasil
1	Himawan & Eliyani (2021)	Perbandingan Akurasi Analisis Sentimen Tweet terhadap Pemerintah Provinsi DKI Jakarta di Masa Pandemi	<i>Random Forest Classifier, Naïve Bayes, Support Vector Machine.</i>	<i>Linear SVM</i> memiliki akurasi terbaik dengan hasil 77,58%, <i>Random Forset Classifier</i> dengan hasil 75,81%, dan <i>Multinomial Naive Bayes</i> sebesar 75,22%..
2	Simorangkir dan Lhaksmana (2018)	Analisis Sentimen pada Twitter untuk Games Online Mobile Legends dan Arena of Valor dengan Metode Naïve Bayes Classifier	<i>Naïve Bayes Classifier</i>	Dapat memprediksi polarisasi sentimen <i>Mobile Legends</i> dengan nilai hasil akurasi, error, recall dan precision yang didapat masing-masing sebesar 88.89%, 19,18%, 96,97%, dan 69,57%. Sedangkan <i>Arena of Valor</i> memiliki nilai

No	Nama Peneliti	Judul Penelitian	Metode	Hasil
				akurasi, eror, recall dan precision masing-masing sebesar 39,02%, 60,98%, 88,89% dan 28,74%.
3	Septian, Fahrudin, dan Nugroho (2019)	Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor	<i>K-Nearest Neighbor</i>	Dari range nilai k=1 hingga k=30 yang merupakan bilangan ganjil, didapatkan akurasi optimal pada k=23 dengan akurasi sebesar 79,99% dan <i>error rate</i> sebesar 20,01%.
4	Pravina, Cholissodin, Adikara (2019)	Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)	<i>Support Vector Machine</i>	Didapatkan nilai parameter <i>learning rate (gamma)</i> sebesar 0,03 dan nilai C sebesar 10 sebagai nilai parameter paling optimal. Didapatkan tingkat akurasi paling baik sebesar 40%, <i>precision</i> sebesar 40%, <i>recall</i> sebesar 100%, dan <i>f-measure</i> sebesar 57,14%.
5	Haranto dan Sari (2019)	Implementasi Support Vector Machine untuk Analisis Sentimen Pengguna Twitter	<i>Support Vector Machine</i>	Menghasilkan nilai <i>accuracy</i> 79,6%, <i>precision</i> 76,5%, <i>recall</i> 72,8% , dan <i>F1-score</i> 74,6% untuk Telkom,

No	Nama Peneliti	Judul Penelitian	Metode	Hasil
		Terhadap Pelayanan Telkom dan Biznet.		serta <i>accuracy</i> 83,2%, <i>precision</i> 78,8%, <i>recall</i> 71,6%, dan <i>F1-score</i> 75% untuk Biznet.
6	Tuhuteru dan Iriani (2018)	Analisis Sentimen Perusahaan Listrik Negara Cabang Ambon Menggunakan Metode Support Vector Machine dan Naive Bayes Classifier	<i>Support Vector Machine</i> dan <i>Naïve Bayes Classifier</i>	Pada penelitian ini, SVM memiliki tingkat akurasi sebesar 81.67%. Sedangkan metode klasifikasi NBC hanya memiliki nilai akurasi sebesar 67.20%.

2.2 Dasar Teori

2.2.1 Twitter

Twitter adalah sebuah situs jejaring media sosial *micro-blogging* gratis yang dikembangkan pada Maret 2006 oleh Jack Dorsey, Noah Glass, Biz Stone, dan Evan Williams dan dapat digunakan oleh khayalak umum semenjak Juli 2006 (Paramastri dan Gumilar, 2019). Media sosial ini memungkinkan *user* untuk menulis pesan singkat yang disebut dengan *tweet*. *Tweet* tersebut bisa berupa teks, video, foto, atau link. Pesan-pesan *tweet* yang ditulis oleh pengguna Twitter tersebut akan ditampilkan di laman *profile*, ditampilkan ke para *followers*, dan juga bisa dicari dengan fitur *search*. (help.twitter.com, 2022).

Laporan terbaru We Are Social di tahun 2022 mengungkapkan suatu fakta bahwa Indonesia adalah salah satu negara dengan pengguna Twitter terbanyak di dunia. Menurut laporan tersebut, jumlah pengguna Twitter di Indonesia pada tahun

2022 mencapai jumlah 18,45 juta pengguna atau setara dengan 4,23% dari total seluruh pengguna Twitter di dunia yang mencapai angka 436 juta (dataindonesia.id, 2022).

2.2.2 Snsrape

Snsrape adalah sebuah *scraper* yang bisa digunakan untuk mengambil data dari *social networking services* (SNS) atau biasa disebut dengan media sosial. Tool ini bisa digunakan untuk mengambil data-data seperti profil pengguna, tagar, pencarian, dan lain-lain pada media sosial seperti Facebook, Instagram, Reddit, Telegram, Twitter, dan lain-lain. Tool ini memerlukan bahasa pemrograman Python versi 3.8 atau lebih supaya bisa diinstall dan digunakan. (JustAnotherArchivist, 2023).

2.2.3 Python

Python adalah bahasa pemrograman *high-level*, interpretatif, multiguna, berorientasi objek dengan semantik dinamis. Sintaks Python yang sederhana dan mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program. Python mendukung modul dan paket, yang mendorong modularitas program dan *code reuse*. *Interpreter* Python dan pustaka standar yang luas tersedia dalam bentuk *source* atau biner tanpa biaya untuk semua platform dan dapat didistribusikan secara bebas. (Python Software Foundation, 2022)

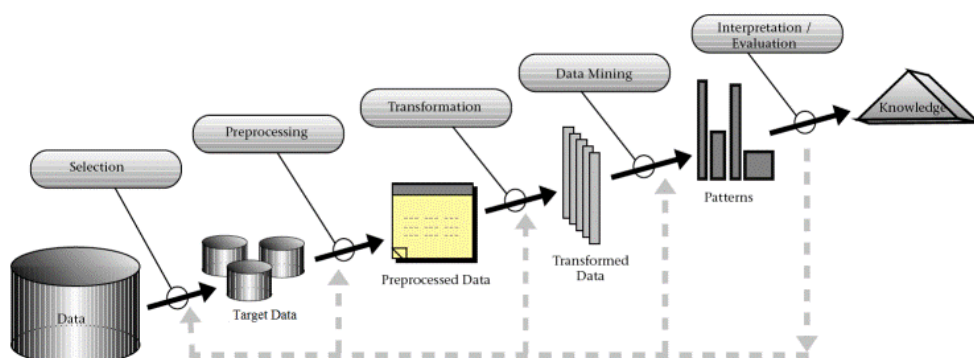
Python membuat penulisan program menjadi padat (*compact*) dan mudah dibaca. Program yang ditulis di Python pada dasarnya memerlukan kode yang lebih sedikit daripada program yang ditulis dengan bahasa C, C++, atau Java karena:

1. Tipe data *high-level* yang dapat melakukan operasi kompleks dalam satu *statement*.
2. Pengelompokan *statement* dilakukan dengan *indentation* (tulisan sedikit menjorok ke kanan), bukan dengan *brackets*.
3. Tidak memerlukan deklarasi variabel atau argumen.

2.2.4 Data Mining

Data mining adalah gabungan dari beberapa ilmu komputer yang didefinisikan sebagai proses penemuan pola-pola baru dari kumpulan data yang sangat besar, meliputi metode-metode yang merupakan bagian dari *artificial intelligence*, *machine learning*, *statistics*, dan *database systems*. Data mining bertujuan untuk mengekstrak pengetahuan dari kumpulan data supaya didapatkan struktur yang dapat dipahami oleh manusia (Suyanto, 2017).

Dalam penerapannya, data mining merupakan salah satu bagian dari sebuah proses yang dinamakan *Knowledge Discovery in Database* (KDD) yaitu proses ekstraksi *non trivial* dari implisit suatu informasi yang sebelumnya tidak diketahui tetapi terdapat potensi informasi yang dihasilkan dari data yang ada (Ependi & Putra, 2019). Grafik KDD ditunjukkan di Gambar 2.1.



Gambar 2.1 Tahapan *Knowledge Discovery Database* (Ependi & Putra, 2019)

Adapun proses *Knowledge Discovery Database* adalah sebagai berikut:

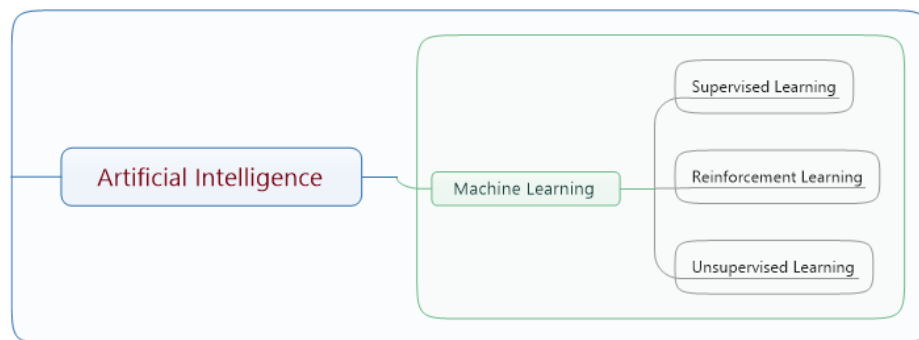
1. *Data Selection*: proses pengambilan data - data yang relevan untuk kemudian dimasukkan ke proses analisis.
2. *Preprocessing*: proses cleaning data yang mencakup beberapa proses seperti membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data.
3. *Data transformation*: proses transformasi dan konsolidasi data ke dalam bentuk yang sesuai untuk ditambang supaya bisa menghasilkan sebuah kesimpulan atau penggabungan.
4. *Data mining*: proses awal yaitu penerapan metode pengkajian untuk mengekstraksi pola data.
5. *Pattern evaluation*: proses mengidentifikasi pola unik yang mewakili basis pengetahuan berdasarkan ukuran tertentu.
6. *Knowledge Presentation*: proses teknik visualisasi dan presentasi yang digunakan untuk menampilkan pengetahuan atau hasil kepada pengguna.

2.2.5 Machine Learning

Machine learning merupakan subbidang dari bidang keilmuan *artificial intelligence*, dengan pemrograman untuk memberikan kecerdasan kepada komputer yang pemahaman maupun kemampuannya dapat ditingkatkan melalui pengalaman secara otomatis. *Machine learning* dapat dilakukan jika ada data yang tersedia sebagai *input* untuk kemudian dilakukan analisis terhadap kumpulan *big data* untuk menemukan pola tertentu. Di dalam *machine learning* dikenal istilah *data training*

dan *data testing*. Proses *data training* digunakan untuk melatih algoritma yang digunakan, sedangkan *data testing* digunakan untuk mengetahui performa dari algoritma *machine learning* yang telah dilatih yaitu ketika diterapkan pada *dataset* baru yang belum pernah diberikan dalam proses *training* (Retnoningsih & Pramudita, 2020).

Machine learning dibagi menjadi tiga kategori yaitu: *supervised learning*, *unsupervised learning*, dan *reinforcement learning* (Roihan, et al., 2020). Grafik relasi antara *artificial intelligence* dan *machine learning* ditunjukkan dalam Gambar 2.2.



Gambar 2.2 Skema *Artificial Intelligence* dan *Machine Learning* (Roihan, et al., 2020)

Supervised learning adalah metode klasifikasi yang memberikan label untuk kumpulan data untuk kemudian diklasifikasikan ke dalam kelas. Sementara pada *unsupervised learning* tidak dibutuhkan pemberian label dalam kumpulan data dan hasilnya tidak mengidentifikasi contoh di kelas yang telah ditentukan. Sedangkan *reinforcement learning* bekerja di dalam lingkungan yang dinamis yang memiliki konsep yaitu harus menyelesaikan tujuan tanpa adanya pemberitahuan dari komputer secara eksplisit jika tujuan tersebut telah tercapai (Roihan, et al., 2020).

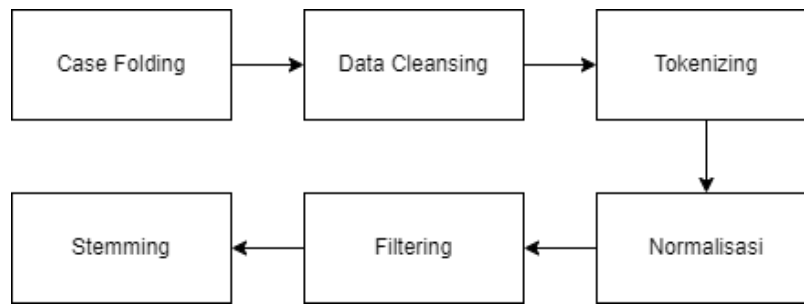
2.2.6 Analisis Sentimen

Analisis sentimen adalah metode komputasional untuk mengekstraksi dan menganalisis sentimen pada suatu entitas dan atribut yang dimiliki (Himawan & Eliyani, 2021). Analisis sentimen dilakukan untuk melihat pendapat terhadap sebuah masalah atau dapat juga digunakan untuk identifikasi kecenderungan dari suatu permasalahan. Tugas dasar dalam analisis sentimen adalah mengelompokkan polaritas dari teks yang ada dalam dokumen, kalimat, atau pendapat. Polaritas mempunyai arti apakah teks yang ada dalam dokumen, kalimat, atau pendapat memiliki aspek positif, netral, atau negatif (Simorangkir & Lhaksana, 2018).

2.2.7 Text Preprocessing

Text Preprocessing atau praproses teks adalah suatu proses yang digunakan untuk melakukan transformasi teks dari yang awalnya berbentuk data tidak terstruktur yang memiliki banyak *noise* menjadi data yang terstruktur sehingga proses analisis sentimen dapat menjadi lebih mudah untuk dilakukan (Husada & Paramita, 2021).

Teks yang berada dalam internet sering mengandung banyak *noise* dan hal mengganggu lainnya seperti tag HTML, *script*, dan iklan. Dengan *preprocessing*, maka *noise* dalam teks dapat dikurangi sehingga hal tersebut bisa meningkatkan performa dan mempercepat proses klasifikasi supaya dapat dengan efektif membantu dalam proses analisis sentimen secara *real-time* (Indrayuni, 2019). *Preprocessing* meliputi beberapa proses yang bisa dilihat pada Gambar 2.3.



Gambar 2.3 Text Preprocessing

Penjelasan tahap *preprocessing* (Septian, et al., 2018):

1. *Case Folding*: proses konversi semua teks dalam suatu dokumen menjadi bentuk yang seragam. Dengan kata lain, *case folding* berfungsi untuk membuat seluruh huruf teks dalam dokumen menjadi huruf kecil.
2. *Data Cleansing*: proses pembersihan pada dokumen yang berisi angka, url (<http://>), username (@), tanda pagar (#), delimiter seperti koma (,) dan titik (.) dan tanda baca lainnya.
3. *Tokenizing*: Proses pemotongan pada dokumen atau kalimat menjadi kata-kata yang disebut dengan token.
4. *Normalisasi*: Proses normalisasi terhadap setiap kata dalam dokumen yang tidak baku menjadi kata yang baku dan siap diolah. Kata tidak baku tersebut artinya adalah kata-kata yang tidak sesuai dengan Kamus Besar Bahasa Indonesia (KBBI).
5. *Filtering*: Proses ini juga bisa disebut dengan *stopword removal*. Proses ini dilakukan untuk menghapus kata-kata yang frekuensi kemunculannya tinggi tetapi tidak berpengaruh terhadap proses analisis data seperti ‘yang’, ‘dan’, ‘ke’, ‘di’, dan lain-lain.

6. *Stemming*: Proses ini merupakan proses untuk mengubah semua kata-kata pada dokumen menjadi kata dasar dengan menghilangkan semua kata imbuhan. Kata imbuhan yang dihilangkan terdiri dari awalan (prefix), akhiran (suffix), sisipan (infix), dan gabungan awalan-akhiran (confix).

2.2.8 Pelabelan Data

Pelabelan data adalah suatu proses untuk menentukan suatu kalimat opini termasuk ke dalam kelas sentimen positif atau sentimen negatif kemudian diberi label sesuai sentimennya. Umumnya, proses pelabelan data membagi kelas menjadi tiga kelas yaitu positif, netral, dan negatif. Skor > 0 akan diklasifikasikan ke dalam kelas sentimen positif, skor < 0 maka akan diklasifikasikan ke dalam kelas negatif, dan skor $= 0$ akan diklasifikasikan ke dalam kelas netral (Mubaroroh et al., 2022).

Proses pelabelan data bisa dilakukan menggunakan *library* Python bernama TextBlob. TextBlob adalah sebuah paket *open-source* pada Python yang berguna untuk melakukan tugas – tugas dasar *Natural Language Processing* seperti tokenisasi, klasifikasi, pelabelan, terjemahan, sentimen analisis, dan lain – lain. (Suanpang et al., 2021). Berikut cara kerja TextBlob:

1. Model yang ada pada TextBlob dilatih dengan memasukkan teks untuk kemudian didapatkan nilai sentimen dalam bentuk polaritas dan subjektivitas.
2. TextBlob memberikan nilai polaritas pada teks masukan. Nilai teks masukan ada di *range* $[-1.0, 1.0]$ dimana skor -1 merupakan teks yang mengandung sentimen negatif dan skor 1.0 merupakan teks yang mengandung sentimen positif.

3. TextBlob juga mendeteksi objektivitas dan subjektivitas pada sebuah teks yang memiliki *range* nilai [0.0, 1.0] dimana nilai 0.0 adalah teks yang sangat objektif, sementara 1.0 adalah teks yang sangat subjektif. Subjektivitas mengukur banyaknya pendapat pribadi dan informasi faktual yang terkandung dalam sebuah teks. Subjektivitas yang tinggi berarti teks tersebut mengandung pendapat pribadi, subjektivitas yang rendah berarti teks tersebut mengandung informasi yang faktual.

2.2.9 Ekstraksi Fitur

Ekstraksi fitur adalah sebuah proses untuk mencari nilai fitur yang terkandung dalam dokumen yang dapat digunakan untuk analisis sentimen atau *opinion mining* (Prihatini, 2017). Proses ini adalah sebuah proses penting pada klasifikasi teks yang digunakan untuk mengubah format tekstual yang tidak terstruktur menjadi format tekstual terstruktur sehingga selanjutnya dapat diproses oleh algoritma *machine learning* untuk diklasifikasikan ke dalam kelas yang telah ditentukan (Budiman, et al., 2020). Ekstraksi fitur bisa dilakukan dengan salah satu *library Python* menggunakan *CountVectorizer*. *CountVectorizer* berfungsi untuk menghitung frekuensi kata dalam suatu dokumen dan juga dapat mengubah fitur teks menjadi representasi *vector* (Munawar & Silitonga, 2019).

N-gram juga diimplementasikan sebagai metode ekstraksi fitur di dalam penelitian ini. Proses ini mengambil sejumlah n karakter sebagai suatu dan menghitung berapa banyak kata itu muncul dan probabilitas dari *n-gram* tersebut. Dengan kata lain, metode ini berguna untuk mengambil potongan – potongan

karakter dari kata atau kalimat sebanyak jumlah karakter pada kata tersebut (Nugroho, 2018).

Dari penjelasan tersebut, dapat dituliskan algoritma atau cara kerja ekstraksi fitur dalam penelitian ini yaitu:

1. Mengubah fitur teks menjadi representasi vektor dengan *CountVectorizer*.

Output dari proses tersebut adalah berupa data dengan tipe *Document Term Matrix* (DTM). DTM adalah suatu matrix yang menggambarkan frekuensi kemunculan kata atau istilah dalam suatu dokumen. Baris pada DTM mempresentasikan dokumen teks dan kolom mempresentasikan istilah teks (Ellina et al., 2022).

2. Contoh sederhana output *CountVectorizer* di Tabel 2.2.

Tabel 2.2 Contoh DTM

	Pernikahan	Di	Usia	Sangat	Muda
DTM-1:					
Pernikahan	1	0	1	0	1
Usia Muda					
DTM-2:					
Pernikahan di	1	1	1	1	1
Usia Sangat					
Muda					

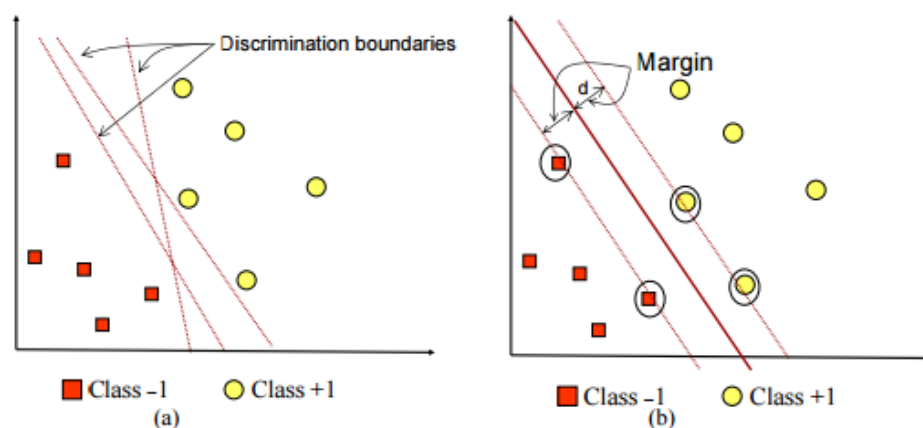
3. Selanjutnya ada proses penerapan *n-gram*. *N-gram* memiliki tiga jenis pemecahan kata yaitu *unigram*, *bigram*, dan *trigram*. *Unigram* adalah pemisahan kata pada teks dengan $n=1$, *bigram* adalah pemisahan kata pada teks dengan $n=2$, *trigram* adalah pemisahan kata pada teks dengan $n=3$ (Anjani dan Fauzan, 2021). Ilustrasi penerapan *n-gram* ada di Gambar 2.3.

Tabel 2.3 Contoh Penerapan *n-gram*

<i>Unigram</i>	‘di’, ‘desa’, ‘saya’, ‘banyak’, ‘yang’, ‘nikah’, ‘muda’
<i>Bigram</i>	‘di desa’, ‘desa saya’, ‘saya banyak’, ‘banyak yang’, ‘yang nikah’, ‘nikah muda’
<i>Trigram</i>	‘di desa saya’, ‘desa saya banyak’, ‘saya banyak yang’, ‘banyak yang nikah’, ‘ yang nikah muda’

2.2.10 Support Vector Machine

Support Vector Machine (SVM) adalah salah satu metode *machine learning* yang bekerja dengan prinsip *Structural Risk Minimization* (SRM) yang termasuk dalam kategori *supervised learning*. Dalam prosesnya, metode SVM memiliki tujuan yaitu untuk menemukan *hyperplane* paling optimal yang berfungsi untuk memisahkan dua buah kelas. Tingkat akurasi pada model yang dihasilkan oleh proses klasifikasi dengan SVM sangat bergantung terhadap fungsi kernel dan parameter yang digunakan. (Parapat, et al., 2018).



Gambar 2.4 Support Vector Machine (Parapat, et al., 2018)

Pada ilustrasi Gambar 2.4, ada dua kelas yang dipisahkan oleh garis *hyperplane* yaitu kelas positif yang bernilai +1 (lingkaran kuning) dan kelas negatif

yang bernilai -1 (kotak merah). Garis solid yang terdapat pada tengah-tengah kedua kelas adalah *hyperplane* terbaik, dan objek merah dan kuning yang berada dalam lingkaran hitam disebut dengan *support vector*.

Pada algoritma *Support Vector Machine*, data ke- i pada dataset diwakilkan dengan variabel x_i , sementara kelas pada dataset diwakilkan dengan variabel y_i . Data x_i yang termasuk dalam kelas +1 dirumuskan dengan persamaan (1), sedangkan data x_i yang termasuk dalam kelas -1 dirumuskan dengan persamaan (2) (Parapat et al., 2018).

$$x_i \cdot w + b \geq 1, y_i = 1 \quad (1)$$

$$x_i \cdot w + b \leq -1, y_i = -1 \quad (2)$$

Keterangan:

x_i = data ke - i

w = nilai bobot *support vector* yang tegak lurus dengan *hyperplane*

b = nilai bias

y_i = kelas data ke- i

Berikut tahap – tahap perhitungan klasifikasi menggunakan SVM:

1. Meminimalkan nilai margin (Zalyhaty et al., 2020). Tahap ini dapat dilakukan dengan menggunakan persamaan berikut:

$$\frac{1}{2} ||w||^2 = \frac{1}{2} (w_1^2 + w_2^2) \quad (3)$$

$$\text{dengan syarat: } y_i(w_i \cdot x_i + b) \geq 1, i = 1, 2, 3, \dots, n \quad (4)$$

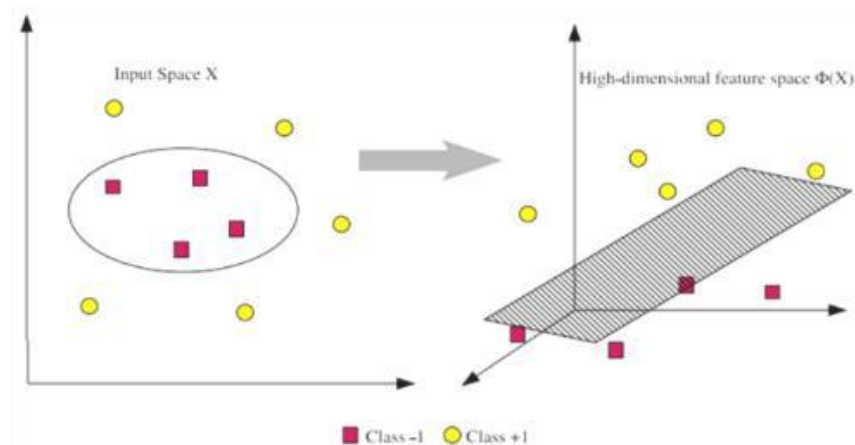
2. Setelah meminimalkan nilai margin, maka dapat ditemukan nilai w (bobot) dan nilai b (bias), lalu bisa dicari persamaan *hyperplane*.

3. Menghitung *margin hyperplane* dalam proses menemukan titik maksimal.

Persamaan (5) adalah rumus untuk memperoleh garis *hyperplane* pada SVM (Husada & Paramita, 2021).

$$w_i \cdot x_i + b = 0 \quad (5)$$

Prinsip kerja algoritma *Support Vector Machine* pada dasarnya adalah suatu algoritma yang digunakan untuk klasifikasi data *linear*, sehingga dalam proses klasifikasi seringkali ditemukan kondisi dimana SVM tidak bekerja dengan baik dalam melakukan klasifikasi pada data *non-linear*. Masalah tersebut bisa diatasi dengan menggunakan *kernel trick*. *Kernel trick* digunakan untuk memetakan data *non-linear* berdimensi rendah ke dalam ruang dimensi yang lebih tinggi sehingga membuat data terpisah secara *linear* lalu dapat terbentuk *hyperplane* yang optimal. Proses klasifikasi dengan SVM dapat dilakukan dengan memilih salah satu di antara 4 kernel yang tersedia yaitu *linear*, *polynomial*, *RBF*, dan *sigmoid* (Husada & Paramita, 2021). Ilustrasi *kernel trick* dapat dilihat di Gambar 2.5.



Gambar 2.5 Pemetaan *Input Space* Berdimensi Dua dengan Pemetaan ke Dimensi Tinggi (Rahutomo et al., 2018)

2.2.11 Evaluasi Performansi

Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (Karsito dan Susanti, 2019). Tabel 2.4 menggambarkan contoh *confusion matrix*.

Tabel 2.4 *Confusion Matrix*

	Kelas Prediksi	
	Positif	Negatif
Positif	True Positive	True Negative
Negatif	False Positive	False Negative

Keterangan isi tabel:

1. *True Positive* (TP), yaitu data asli positif dan data klasifikasi positif.
2. *True Negative* (TN), yaitu data asli negatif dan data klasifikasi negatif.
3. *False Positive* (FP), yaitu data asli negatif dan data klasifikasi positif.
4. *False Negative* (FN), yaitu data asli positif dan data klasifikasi negatif.

Setelah itu dapat dilakukan perhitungan untuk menghasilkan accuracy, precision, recall, dan f1-score. *Accuracy* adalah perbandingan kasus yang diidentifikasi benar dengan jumlah semua data. *Precision* adalah rasio prediksi benar positif dibandingkan dengan hasil prediksi positif secara keseluruhan. *Recall* adalah rasio benar positif dibandingkan dengan seluruh data positif. *F1-Score*

adalah parameter perbandingan rata-rata *precision* dan *recall* yang dibobotkan (Hidayat, Ardiansyah, & Setyanto, 2021). Berikut rumusnya:

$$\text{Akurasi} = \frac{TP+TN}{TP+FN+FN+TN} \times 100 \% \quad (6)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

BAB 3

METODE PENELITIAN

3.1 Bahan/Data

Gambaran atau deskripsi analisis kebutuhan dalam penelitian ini.

3.1.1 Kebutuhan Input

Data yang digunakan dalam penelitian ini adalah tanggapan masyarakat di Indonesia mengenai topik pernikahan di usia muda yang berupa data *tweets* yang diambil dari media sosial Twitter. Data yang digunakan dalam penelitian ini diambil menggunakan suatu kata kunci yaitu “nikah muda”. Data yang diambil berjumlah 2000 data yang diambil pada bulan Maret 2023, dan 2000 data yang diambil pada bulan April 2023. Total data berjumlah 4000 data *tweet*. Data tersebut nantinya akan dibagi menjadi dua jenis data yaitu data latih dan data uji. Data latih diambil sebanyak 80% dan data uji sebanyak 20% dari data keseluruhan.

3.1.2 Kebutuhan Proses

Penelitian ini diawali dengan proses pengambilan data *tweet* dari Twitter dengan proses yang disebut dengan *scraping*. Data yang telah diambil dari Twitter tersebut akan melewati proses yang disebut dengan *preprocessing* untuk mengubah data mentah menjadi data yang siap digunakan serta memastikan kualitas data sudah cukup baik untuk diproses saat analisis data. Tahap *preprocessing* terdiri dari *case folding*, *data cleansing*, *tokenizing*, *normalisasi*, *filtering*, dan *stemming*.

Data yang telah melewati tahap *preprocessing* tersebut kemudian akan melalui proses pelabelan data dengan *TextBlob*, kemudian ekstraksi fitur dengan *CountVectorizer* dan *n-gram*. Tahap selanjutnya pelatihan dan pengujian data

menggunakan metode Support Vector Machine. Terakhir adalah implementasi model *Support Vector Machine* yang digunakan untuk klasifikasi sentimen positif, netral, dan negatif.

3.1.3 Kebutuhan Output

Analisis sentimen pada penelitian ini akan menghasilkan output berupa hasil klasifikasi teks bersentimen positif, netral, dan negatif.

3.2 Peralatan

Peralatan dalam penelitian ini adalah kebutuhan perangkat lunak dan kebutuhan perangkat keras untuk mendukung penelitian.

3.2.1 Kebutuhan Perangkat Lunak

Perangkat lunak (*software*) adalah program, *tool*, bahasa pemrograman, dan sistem operasi yang digunakan pada penelitian ini. Berikut beberapa perangkat lunak yang digunakan:

1. Sistem operasi : Windows 10 Pro-64 bit (10.0, Build 19043)
2. Bahasa pemrograman : Python 3.9.7
3. IDE : Jupyter Notebook dan Visual Studio Code
4. *Tools* desain: Diagrams.net dan CorelDraw X7

3.2.2 Kebutuhan Perangkat Keras

Perangkat keras (*hardware*) adalah bagian fisik pada komputer yang digunakan untuk membuat dan menjalankan sistem pada penelitian ini. Berikut perangkat keras yang digunakan:

1. Laptop Lenovo Ideapad 330-14AST, dengan spesifikasi:
 - a. Processor AMD A4-9125 Radeon R3 2.3 GHz

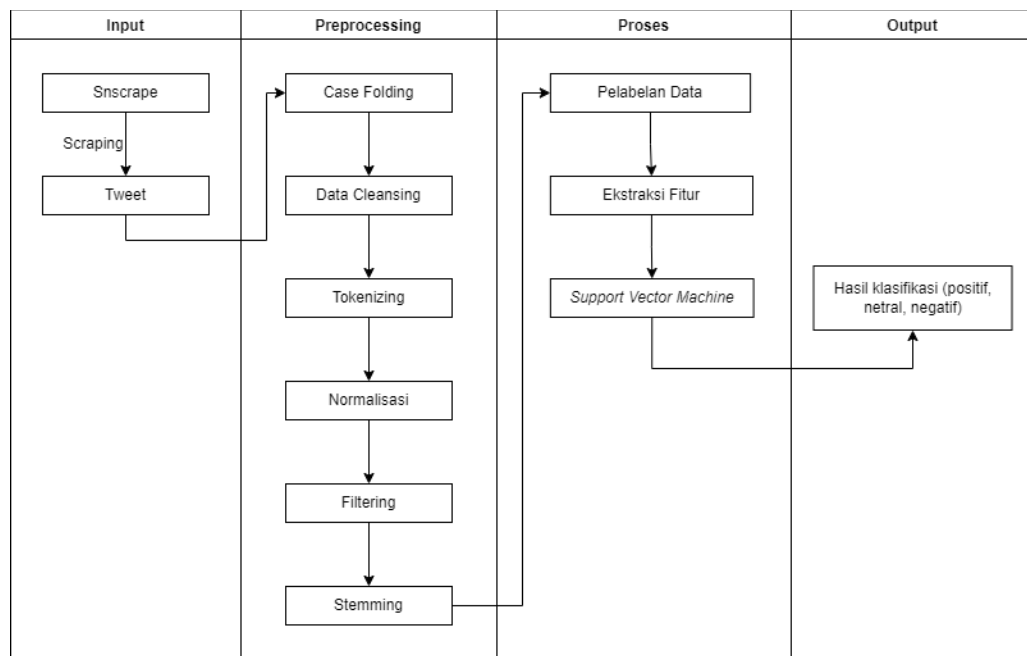
- b. Random Access Memory (RAM) 8 GB
- c. Penyimpanan HDD 500 GB
- d. VGA card AMD Radeon™ R3 Graphics

3.3 Prosedur Pengumpulan Data

Metode pengumpulan data dilakukan melalui proses *data scraping* yang dilakukan dengan *scraping tool* bernama Snscape dengan mekanisme pencarian berdasarkan keyword “nikah muda”. Teknik *data scraping* dilakukan dengan menggunakan bahasa pemrograman Python dengan meng-*import library* Snscape. Dengan library tersebut, kita menggunakan fungsi *TwitterSearchScrapper* untuk mendapatkan *tweet* sesuai dengan *query* yang kita masukkan.

3.4 Analisis dan Rancangan Sistem

3.4.1 Block Diagram



Gambar 3.1 Block Diagram

Block diagram pada Gambar 3.1 menggambarkan alur yang akan dijalankan yaitu tahap *input*, proses, dan *output* yang berjalan selama analisis sentimen dilakukan.

Penjelasan setiap tahap:

1. *Input* dilakukan dengan mengambil dan mengumpulkan data tweet dengan *scraping* menggunakan *tool* Snsrape.
2. Tahap *pre-processing* yang terdiri dari beberapa tahap yaitu:
 - a. *Case folding*: proses untuk mengubah semua karakter alfabet menjadi huruf kecil atau *lowercase*.
 - b. *Data Cleansing*: penghapusan dokumen yang memiliki angka, url (<http://>), username (@), tanda pagar (#), koma (,), titik (.) dan berbagai tanda baca lainnya.
 - c. *Tokenizing*: memecah teks dokumen menjadi kalimat, kemudian memecahnya lagi menjadi kata-kata yang disebut dengan token.
 - d. Normalisasi: mengubah kata-kata yang tidak baku menjadi baku. Contohnya kata “ga” menjadi “tidak”. Di proses ini juga mengubah bahasa daerah menjadi bahasa Indonesia, contohnya “ora” menjadi “tidak”.
 - e. *Filtering*: menghapus kata-kata yang frekuensi kemunculannya tinggi tapi tidak penting seperti ”yang”, ”dan”, ”ke”, ”di”. Di proses ini juga menghapus kata-kata yang tidak memiliki arti atau tidak berpengaruh seperti “ckckckck”

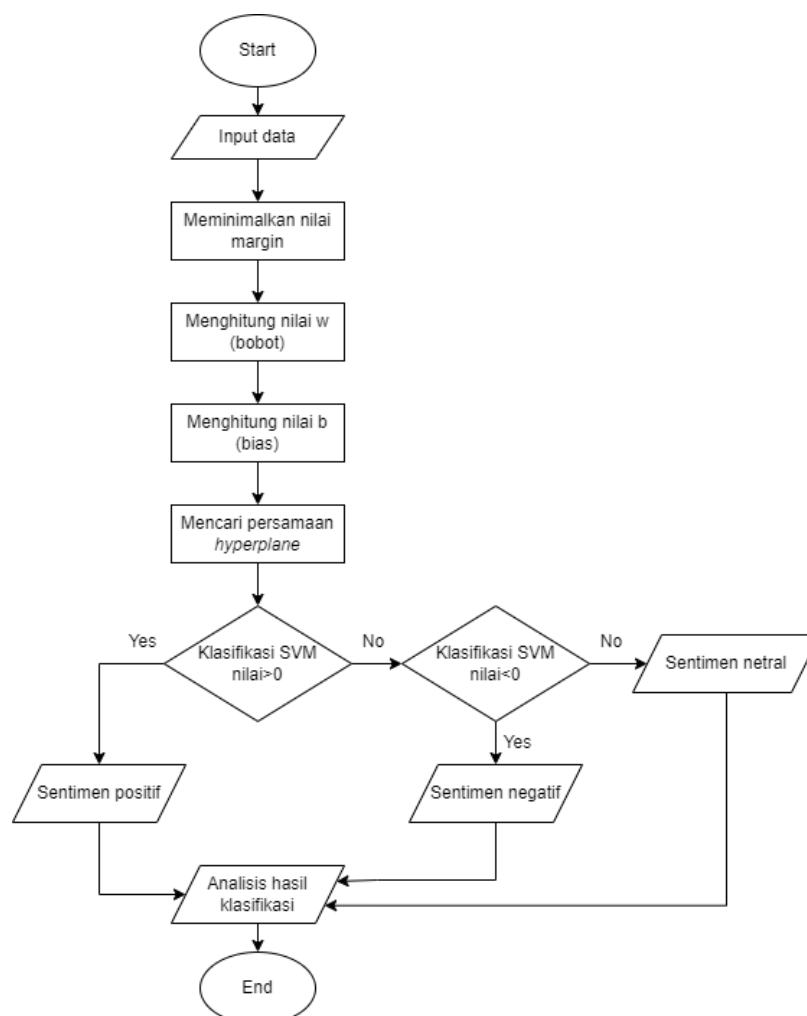
- f. *Stemming*: mencari kata dasar setiap kata dalam dokumen dengan membuang imbuhan awal maupun imbuhan akhir. Contohnya “persiapan” menjadi “siap”.
3. Setelah melalui tahap *preprocessing*, proses selanjutnya adalah memberi label positif, netral, negatif pada data *tweet*. Sebelum dilakukan pelabelan data, dataset hasil *preprocessing* tersebut diterjemahkan terlebih dahulu ke dalam bahasa Inggris menggunakan Google Spreadsheet. Dataset harus diterjemahkan terlebih dahulu karena TextBlob hanya bisa mendeteksi bahasa Inggris. Pada penelitian ini digunakan library Python yaitu TextBlob untuk memberi label positif, netral, dan negatif sesuai dengan polaritas data *tweet*.
4. Semua data yang telah melalui tahap *labeling* akan melalui proses ekstraksi fitur supaya data bisa dibaca oleh model *machine learning*. Pada pencarian fitur ini, dilakukan dua proses yaitu pembuatan *word vector* yaitu proses mengubah fitur teks menjadi visualisasi *vector* dengan parameter *n-gram* untuk menentukan pemecahan kata sesuai jumlah *n* yang ditentukan. Setelah melalui proses itu, data telah siap digunakan untuk data *training* dan akan melewati proses algoritma *Support Vector Machine*.
5. Selanjutnya adalah pelatihan dan pengujian data dengan metode *Support Vector Machine*. Dengan metode ini, garis pembatas atau *hyperplane* digunakan untuk memisahkan tiga kelas sentimen yaitu data *tweet* positif, netral, dan negatif pada 4000 data yang sudah melewati ekstraksi fitur. Performa model algoritma tersebut kemudian akan diuji menggunakan

confusion matrix supaya kemudian bisa dihitung *accuracy*, *precision*, *recall*, dan *f1-score*.

6. Model machine learning tersebut kemudian bisa digunakan untuk menentukan untuk mengklasifikasikan data ke dalam tiga kelas yaitu positif, netral, dan negatif.

3.4.2 Flowchart Support Vector Machine

Flowchart atau diagram alir di Gambar 3.2 menggambarkan algoritma klasifikasi Support Vector Machine.



Gambar 3.2 Flowchart SVM

Langkah - langkah:

1. Langkah pertama dari klasifikasi dengan algoritma *Support Vector Machine* adalah memasukkan data.

Tabel 3.1 Tabel Contoh Masukan Data Klasifikasi SVM

ID Data	x_1	x_2	Kelas (y)
D1	1	1	1
D2	1	-1	-1
D3	-1	1	-1
D4	-1	-1	-1

Pada Tabel 3.1, terdapat dua fitur yaitu x_1 dan x_2 sehingga akan ada dua bobot yaitu w_1 dan w_2 .

2. Meminimalkan nilai margin dengan rumus:

$$\frac{1}{2} ||w||^2 = \frac{1}{2} (w_1^2 + w_2^2)$$

dengan syarat: $y_i(w_i \cdot x_i + b) \geq 1, i = 1, 2, 3, \dots, n$

3. Masukkan data ke dalam rumus syarat seperti berikut:

$$y_i(w_1 \cdot x_1 + w_2 \cdot x_2 + b) \geq 1$$

$$\text{Persamaan 1: } 1(w_1 \cdot 1 + w_2 \cdot 1 + b) \geq 1 \quad \Rightarrow w_1 + w_2 + b \geq 1$$

$$\text{Persamaan 2: } -1(w_1 \cdot 1 + w_2 \cdot (-1) + b) \geq 1 \quad \Rightarrow -w_1 + w_2 - b \geq 1$$

$$\text{Persamaan 3: } -1(w_1 \cdot (-1) + w_2 \cdot 1 + b) \geq 1 \quad \Rightarrow w_1 - w_2 - b \geq 1$$

$$\text{Persamaan 4: } -1(w_1 \cdot (-1) + w_2 \cdot (-1) + b) \geq 1 \quad \Rightarrow w_1 + w_2 - b \geq 1$$

4. Menjumlahkan persamaan untuk menemukan nilai w_1 , w_2 , dan b

Menjumlahkan persamaan 1 dan 2:

$$w_1 + w_2 + b \geq 1$$

$$\begin{array}{rcl}
 -w_1 + w_2 - b & \geq & 1 \\
 \hline
 2w_2 & = & 2 \\
 w_2 & = & 1
 \end{array}$$

Menjumlahkan persamaan 1 dan 3:

$$\begin{array}{rcl}
 w_1 + w_2 + b & \geq & 1 \\
 w_1 - w_2 - b & \geq & 1 \\
 \hline
 2w_1 & = & 2 \\
 w_1 & = & 1
 \end{array}$$

Menjumlahkan persamaan 2 dan 3:

$$\begin{array}{rcl}
 -w_1 + w_2 - b & \geq & 1 \\
 w_1 - w_2 - b & \geq & 1 \\
 \hline
 -2b & = & 2 \\
 b & = & -1
 \end{array}$$

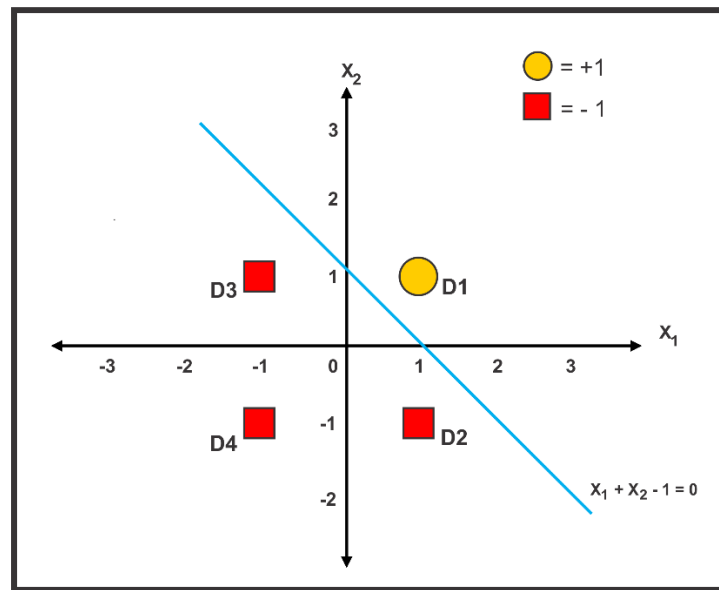
5. Sehingga diketahui persamaan *hyperplane*

$$\begin{aligned}
 w_1 \cdot x_1 + w_2 \cdot x_2 + b &= 0 & \Rightarrow & 1 \cdot x_1 + 1 \cdot x_2 + (-1) = 0 \\
 & & \Rightarrow & x_1 + x_2 - 1 = 0
 \end{aligned}$$

6. Mencari titik potong dengan cara substitusi salah satu x menjadi 0:

$$\begin{aligned}
 0 + x_2 - 1 &= 0 & \Rightarrow & x_2 = 1 \\
 x_1 + 0 - 1 &= 0 & \Rightarrow & x_1 = 1
 \end{aligned}$$

7. Dari perhitungan tersebut, bisa diketahui jika garis *hyperplane* akan melewati titik (x_1, x_2) atau $(1, 1)$. Visualisasi *hyperplane*:



Gambar 3.3 Visualisasi *Hyperplane*

Berdasarkan Gambar 3.3, bisa disimpulkan bahwa data yang berada di atas garis *hyperplane* masuk dalam kelas positif, sementara data yang berada di bawah garis *hyperplane* masuk dalam kelas negatif.

Tabel 3.2 Tabel Hasil Analisis Klasifikasi SVM

ID Data	Kelas
D1	Positif
D2	Negatif
D3	Negatif
D4	Negatif

Pada penelitian ini, digunakan kernel *Linear* karena kernel *Linear* memiliki akurasi dan performa yang paling baik dibandingkan dengan kernel *Polynomial* dan RBF dalam mengklasifikasikan 4000 data yang digunakan pada penelitian ini.

3.4.3 Perancangan Antarmuka

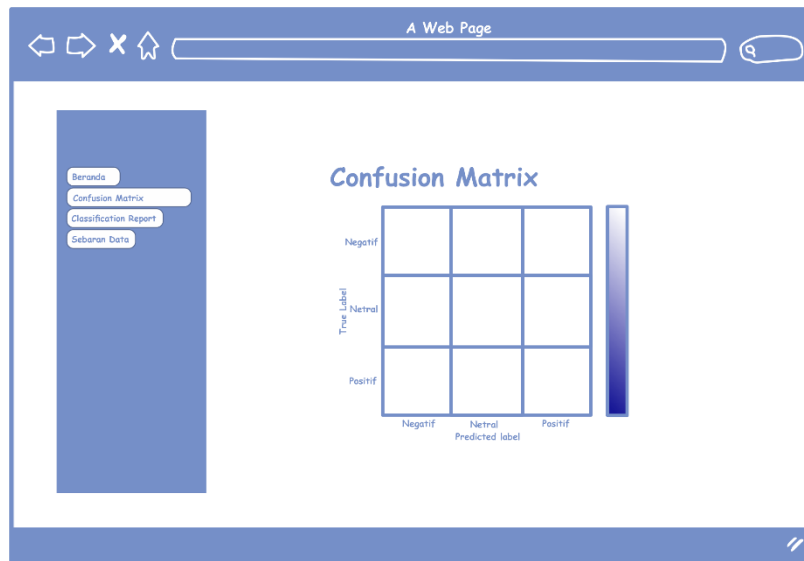
Antarmuka atau dalam bahasa inggris bisa disebut dengan *interface* adalah sebuah fitur yang dimiliki oleh sebuah aplikasi, website, atau sistem yang

digunakan sebagai sarana interaksi antara pengguna dengan sistem. Penelitian ini menggunakan Streamlit sebagai *framework* untuk membangun aplikasi web *machine learning*.



Gambar 3.4 Desain Halaman Beranda

Gambar 3.4 merupakan halaman utama atau beranda aplikasi web. Di halaman ini disediakan sebuah *text field* untuk memasukkan teks yang akan diklasifikasi. Ada tombol “Klasifikasi sentimen teks” untuk memulai proses klasifikasi teks yang dimasukkan di dalam *text field*. Di samping kiri aplikasi web, tersedia *Sidebar* berisi semua menu atau halaman yaitu menu “Beranda”, menu “Confusion Matrix”, menu “Classification Report”, dan menu “Sebaran Data”.



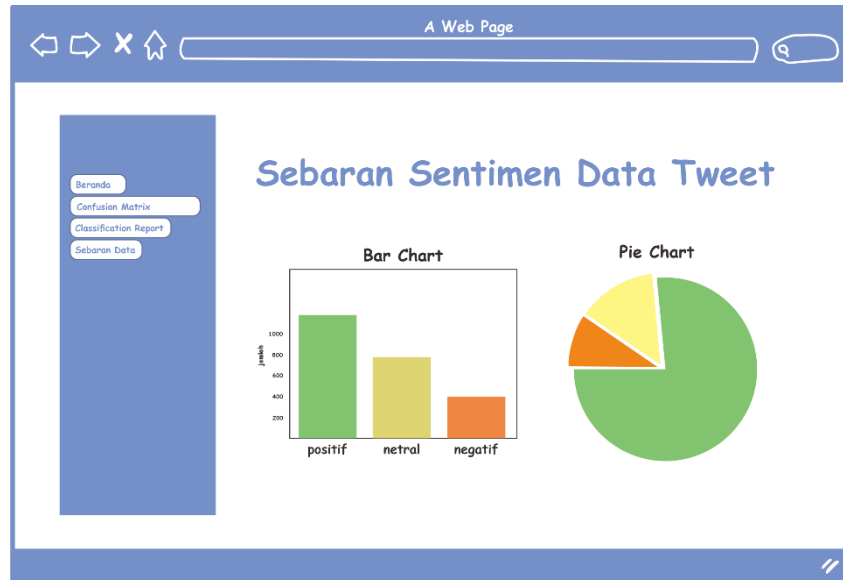
Gambar 3.5 Desain Halaman Confusion Matrix

Halaman “Confusion Matrix” pada Gambar 3.5 berisi diagram *confusion matrix* *Support Vector Machine*, sementara halaman “Classification Report” pada Gambar 3.6 berisi nilai *precision*, *recall*, dan *F1-score* setiap kelas.

	precision	recall	f1-score	support
negatif	0,00	0,00	0,00	300
netral	0,00	0,00	0,00	200
positif	0,00	0,00	0,00	300
accuracy				
micro avg	0,00	0,00	0,00	0,00
weighted avg	0,00	0,00	0,00	0,00

Gambar 3.6 Desain Halaman Classification Report

Terakhir ada halaman “Sebaran Data” pada Gambar 3.7 yang berisi *bar chart* dan *pie chart* untuk menghitung banyaknya sebaran sentiment data *tweet*.



Gambar 3.7 Desain Halaman Sebaran Data

BAB 4

IMPLEMENTASI DAN PEMBAHASAN SISTEM

4.1 Implementasi

Tahap implementasi merupakan tahap yang menguraikan tentang pembangunan dan penerapan sistem sesuai dengan rancangan yang sebelumnya telah dibuat.

4.1.1 Pengambilan Data

Data berjumlah 4000 *tweet* dengan keyword “nikah muda” berbahasa Indonesia yang diambil dari Twitter pada bulan Maret dan April 2023 menggunakan Snsrape dan disimpan dalam bentuk file *xlsx*. Kode ditunjukkan di Gambar 4.1 dan Gambar 4.2.

```
query = "nikah muda lang:id since:2023-03-01 until:2023-04-01"
tweets = []

for i, tweet in
enumerate(sntwitter.TwitterSearchScrapper(query).get_items()):
    if i >= 2000:
        break
    else:
        tweets.append([tweet.date, tweet.user.username, tweet.content])

maret = pd.DataFrame(tweets, columns=['date', 'username', 'tweet'])
maret['date'] = maret['date'].dt.tz_localize(None)
maret.to_excel('maret.xlsx', index=False)
```

Gambar 4.1 Kode *Scraping* Bulan Maret

```
query = "nikah muda lang:id since:2023-04-01 until:2023-05-01"
tweets = []

for i, tweet in
enumerate(sntwitter.TwitterSearchScrapper(query).get_items()):
    if i >= 2000:
        break
    else:
        tweets.append([tweet.date, tweet.user.username, tweet.content])

april = pd.DataFrame(tweets, columns=['date', 'username', 'tweet'])
april['date'] = april['date'].dt.tz_localize(None)
april.to_excel('april.xlsx', index=False)
```

Gambar 4.2 Kode *Scraping* Bulan April

4.1.2 Preprocessing

Adalah tahap untuk mengubah data mentah menjadi data yang siap digunakan supaya memudahkan dalam proses analisis sentimen. Tahap ini perlu dilakukan karena beberapa kalimat *tweet* yang didapatkan tidak sepenuhnya menggunakan kata baku dan tidak menggunakan bahasa yang baik dan benar.

1. *Case Folding*

Proses mengubah data teks menjadi *lowercase*. Kode bisa dilihat di Gambar 4.3 dan contoh bisa dilihat di Tabel 4.1.

```
df['case folding'] = df['tweet'].str.lower()
```

Gambar 4.3 Kode *Case Folding*

Tabel 4.1 Tabel *Case Folding*

Sebelum	Sesudah
@AREAJULID drpd nikah muda tp ngajak anak orang susah:(@areajulid drpd nikah muda tp ngajak anak orang susah:(

2. *Data Cleansing*

Memodifikasi, mengubah, atau menghapus data-data yang dianggap tidak perlu. Tahap ini menghilangkan angka, karakter spesial, tanda baca, *link*, *hashtag*, *emoji*, dan lain-lain. Kode bisa dilihat di Gambar 4.4 dan contoh bisa dilihat di Tabel 4.2.

```

def remove_tweet_special(text):
    # remove tab, new line, ans back slice
    text = text.replace('\t'," ").replace('\n',"
").replace('\u'," ").replace('\',"")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9+])|(\w+:\/\/\/\S+)", "
", text).split())
    # remove incomplete URL
    return text.replace("http://", " ").replace("https://", "
")
df['data cleansing'] = df['case
folding'].apply(remove_tweet_special)

#remove number
def remove_number(text):
    return re.sub(r"\d+", "", text)

df['data cleansing'] = df['data
cleansing'].apply(remove_number)

#remove punctuation
def remove_punctuation(text):
    return
text.translate(str.maketrans("", "", string.punctuation))

df['data cleansing'] = df['data
cleansing'].apply(remove_punctuation)

#remove whitespace leading & trailing
def remove_whitespace_LT(text):
    return text.strip()

df['data cleansing'] = df['data
cleansing'].apply(remove_whitespace_LT)

#remove multiple whitespace into single whitespace
def remove_whitespace_multiple(text):
    return re.sub('\s+', ' ',text)

df['data cleansing'] = df['data
cleansing'].apply(remove_whitespace_multiple)

# remove single char
def remove_single_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", text)

df['data cleansing'] = df['data
cleansing'].apply(remove_single_char)

```

Gambar 4.4 Kode Data Cleansing

Tabel 4.2 Tabel Data Cleansing

Sebelum	Sesudah
@areajulid drpd nikah muda tp ngajak anak orang susah:(drpd nikah muda tp ngajak anak orang susah

3. *Tokenizing*

Tokenizing dalam penelitian ini merupakan tahapan dalam memecah *string* pada suatu teks yang telah melewati tahap *data cleansing* menjadi pecahan kata-kata yang disebut dengan token. Kode bisa dilihat di Gambar 4.5 dan contoh bisa dilihat di Tabel 4.3.

```
def tokenization(text):
    return word_tokenize(text)

df['tokenization'] = df['data cleansing'].apply(tokenization)
```

Gambar 4.5 Kode *Tokenizing*

Tabel 4.3 Tabel *Tokenizing*

Sebelum	Sesudah
drpd nikah muda tp ngajak anak orang susah	['drpd', 'nikah', 'muda', 'tp', 'ngajak', 'anak', 'orang', 'susah']

4. Normalisasi

Normalisasi adalah tahap untuk mengubah kata tidak baku atau *slang words* menjadi kata baku. Pada tahap ini, terlebih dahulu dibuat secara manual sebuah file bernama “normalisasi.xlsx” yang berisi daftar bahasa tidak baku dan bahasa bakunya dengan file *xlsx* yang bisa dilihat di Gambar 4.6.

Pada kode program normalisasi ini, *row* pertama merupakan bahasa tidak baku, *row* kedua merupakan bahasa baku. Kode program di Gambar 4.7 akan menggantikan semua kata di *row* pertama dengan semua kata di *row* kedua. Contoh hasil penerapan normalisasi ada di Tabel 4.4.



	A	B	C	D
839	bgt	banget		
840	org	orang		
841	jg	juga		
842	brengsek	berengsek		
843	dr	dari		
844	pd	pada		
845	ntar	nanti		
846	lgsg	langsung		
847	lo	kamu		
848	ni	ini		
849	nyinyir	menyindir		
850	join	gabung		
851	nikahnika	nikah		
852	gw	aku		
853	sm	sama		
854	masi	masih		
855	cri	cari		
856	bpak	bapak		
857	skrg	sekarang		
858	udh	sudah		
859	th	tahun		
860	mo	mau		

Gambar 4.6 File Normalisasi

```
normalized_word = pd.read_excel("normalisasi.xlsx")

normalized_word_dict = {}

for index, row in normalized_word.iterrows():
    if row[0] not in normalized_word_dict:
        normalized_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalized_word_dict[term] if term in
            normalized_word_dict else term for term in document]

df['normalisasi'] = df['tokenization'].apply(normalized_term)
```

Gambar 4.7 Kode Normalisasi

Tabel 4.4 Tabel Normalisasi

Sebelum	Sesudah
['drpd', 'nikah', 'muda', 'tp', 'ngajak', 'anak', 'orang', 'susah']	['daripada', 'nikah', 'muda', 'tapi', 'mengajak', 'anak', 'orang', 'susah']

5. Filtering

Pada tahap *filtering* ini, dilakukan *stopwords removal* yaitu penghapusan kata yang tidak memiliki arti dan frekuensinya banyak.

```
# get stopwords indonesia
list_stopwords = stopwords.words('indonesian')

# ----- manually add stopwords -----
# append additional stopwords
list_stopwords.extend(['nik', 'ais', 'ih', 'kuea', 'ndes', 'tk', 'arg
hhhh', 'wuakakak', 'gtth',
'wowww', 'apeeee', 'Aksjsjsk', 'alaae', 'koq', 'wuakakak', 'salengpra
ew', 'rukhhadevata', 'gtth', 'zeon', 'vivienne', 'yaam', 'woyy', 'ykwi
m', 'auff', 'ue', 'hoek', 'hayo', 'chnmn', 'hahahah', 'haaaaaa', 'din',
'woy', 'ndeer', 'lalalala', 'wkwkwkwkwkwk', 'woyyy', 'dih', 'den', 'he
hehew', 'etdah', 'beeeuh', 'wahh', 'heheee', 'hhaaha', 'waaaaa', 'oaki
lah', 'haaaahh', 'huft', 'ai', 'et', 'acha', 'ue', 'hokyahokya', 'hahah
ihi', 'yl', 'wihh', 'hahahaa', 'hhhh', 'def', 'ayom', 'ser', 'duh', 'heu
heueheu', 'huwaaaaaa', 'yalah', 'mww', 'cekabia', 'dikatar', 'anggan
ara', 'krtsk', 'woee', 'ndi', 'ohh', 'www', 'aee', 'huaaaa', 'gn', 'haha
hah', 'nd', 'ema', 'ceratops', 'pasuk', 'ygy', 'repp', 'gais', 'hadehhh
hhhh', 'walah', 'hahah', 'paa', 'awkwkwk', 'wkwkk', 'wkwkw', 'wkwkwkwk
wkwah', 'wkwkwkw', 'baceprot', 'sksksk', 'heheh', 'brooo', 'dbd', 'aee
e', 'weeeh', 'wehh', 'milita', 'hsnah', 'swsg', 'hemm', 'xda', 'yara',
'ohh', 'heh', 'kle', 'acy', 'hayooo', 'hahahahaha', 'balablablabla', '
lai', 'loj', 'itine', 'hehehehe', 'kwkwk', 'kwkwkwkwkwkwk', 'waaa', 'dem
ending', 'pali', 'eeh', 'dlsb', 'cooooy', 'hehehehe', 'adjem', 'aih', '
syar', 'wkwkk', 'aowkwkwk', 'walah', 'euy', 'der', 'hahaa', 'hesteg', '
hmmmmmtar', 'gtideologi', 'ab', 'owkwkwkwk', 'dncw', 'sloga', 'jo', 'je
ngjenggg', 'anuanu', 'caw', 'ehheheheh', 'hlaa', 'hahahihi', 'ckckckc
k', 'sich', 'pakin', 'mmarkpkk', 'ponponpon', 'kyary', 'pamyu', 'laaah
hh', 'cp', 'duhhh', 'napen', 'lise', 'bi', 'ieu', 'poho', 'boga', 'imah',
'keur', 'ulin', 'kwkwkw', 'ehheh', 'gryli', 'oalah', 'prekk', 'hehh',
'cere', 'ekekekek', 'chco', 'nganu', 'wkwkwkwkwkwkwk', 'pfft', 'awowk
wkwkwk', 'kinyis', 'pus', 'yng', 'yg', 'yang', 'wkwoswkw', 'wkwkwkwkw
kwk', 'ahahha', 'weeeeh', 'hah', 'fir', 'hong', 'jay', 'haikyuu',
'nderrr', 'omtanteuwaksodara', 'ahsajkakaka', 'kwkwkwk', 'derrr',
'wkwkwkwk', 'hadehh', 'aaaaa', 'heeh', 'dem', 'ocaaa', 'wo', 'prenup',
'dihhh', 'cokk', 'imho', 'chenle', 'jsdieksisawikwok', 'hahahahah
ahaha', 'bam', 'yowohh', 'lau', 'boiiiii', 'gih', 'beuhhh', 'wkw', 'wkw
kwkw', 'dooong', 'oalaaaa', 'sinoeng', 'wkekwk', 'nyai', 'cai', 'anw',
'tjuyyy', 'hanss', 'mh', 'ih', 'widihh', 'cy', 'eeee', 'gi', 'luat', '
laaaaa', 'cam', 'lancau', 'tuch', 'kun', 'uhhhh', 'chuakssss', 'oiyaa',
'hadeuhhhh', 'wkwkwkwkwk', 'heheheh', 'nk', 'lak', 'qwq', 'oneesan', '
eeehmmm', 'am', 'wkwk'])

# ----- add stopwords from txt file -----
txt_stopword = pd.read_csv("stopwordbahasa.txt", names=
["stopwords"], header = None)

list_stopwords.extend(txt_stopword["stopwords"][0].split(' '))
list_stopwords = set(list_stopwords)

def stopwords_removal(words):
    return [word for word in words if word not in
list_stopwords]

df['stopwords'] = df['normalisasi'].apply(stopwords_removal)
```

Gambar 4.8 Kode *Filtering*

Tabel 4.5 Tabel *Filtering*

Sebelum	Sesudah
['daripada', 'nikah', 'muda', 'tapi', 'mengajak', 'anak', 'orang', 'susah']	['nikah', 'muda', 'mengajak', 'anak', 'orang', 'susah']

6. *Stemming*

Tahap *preprocessing* selanjutnya adalah *stemming* yaitu tahap mencari *root* (dasar) kata dari tiap kata hasil filtering dengan menghapus kata imbuhan di depan maupun imbuhan di belakang kata. Kode bisa dilihat di Gambar 4.9 dan contoh di tabel 4.6.

```
# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmer
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in df['stopwords']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("-----")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term, ":", term_dict[term])

print(term_dict)
print("-----")

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df['stemmed'] = df['stopwords'].swifter.apply(get_stemmed_term)
```

Gambar 4.9 Kode *Stemming*

Tabel 4.6 Tabel Stemming

Sebelum	Sesudah
['nikah', 'muda', 'mengajak', 'anak', 'orang', 'susah']	['nikah', 'muda', 'ajak', 'anak', 'orang', 'susah']

7. Penggabungan Kata

Menggabungkan kata-kata yang sudah berbentuk token menjadi kalimat kembali untuk mempermudah proses penerjemahan ke dalam bahasa inggris. Kode bisa dilihat di Gambar 4.10 dan contoh di Tabel 4.7.

```
def fit_stopwords(text):
    text= np.array(text)
    text= ' '.join(text)
    return text

df['text']=df['stemmed'].apply(lambda x: fit_stopwords(x))
```

Gambar 4.10 Kode Penggabungan Kata**Tabel 4.7 Tabel Penggabungan Kata**

Sebelum	Sesudah
['nikah', 'muda', 'ajak', 'anak', 'orang', 'susah']	nikah muda ajak anak orang susah

4.1.3 Labeling

Pelabelan data dilakukan dengan menggunakan library python TextBlob dengan melihat *polarity* yang dimiliki oleh teks *tweet* yang telah diterjemahkan ke dalam bahasa inggris menggunakan fungsi GOOGLETRANS milik Google Spreadsheet. TextBlob saat ini hanya menyediakan layanan *labeling* untuk data berbahasa inggris sehingga dataset harus diterjemahkan ke dalam bahasa inggris terlebih dahulu. Proses penerjemahan ditunjukkan di Gambar 4.11.

12:11 | fx =GOOGLETRANSLATE(H2;"id";"en")

	A	B	C	D	E	F	G	H	I	J	K
1	tweet	ase foldin	ta cleansip	kenizatio	normalisat	stopwords	stemmed	text	english		
2	Ap lgi nika	ap lgi nika									
3	Buset udh	buset udh									
4	Pokok ny	pokok ny	ap lgi nika	['ap', 'lgi',	['apa', 'lag	['nikah', 'i	['nikah', 'i	nikah mu	Marriage	Young Marriage E	
5	@lalalaaa	@lalalaaa									
6	Aku punya	aku punya	kurang va	['kurang',	['kurang',	['valid', 'n	['valid', 'n	valid mbk	Valid mbk	grandfather my i	
7	@Fenggz	@fenggz	perempui	['perempi	['perempi	['perempi	['perempi	perempui	beautiful	young women stu	
8	Tiap buka	tiap buka	tiap buka	['tiap', 'bu	['tiap', 'bu	['buka', 'ti	['buka', 'ti	buka tele	Open the	television filled v	
9	Minimo b	minimo b									

Gambar 4.11 Google Spreadsheet

Penentuan kelas positif, netral dan negatif didasari oleh nilai polaritas. Teks *tweet* dengan nilai polaritas mengarah ke nilai 1 menunjukkan sentimen kelas positif, nilai polaritas mengarah ke nilai -1 menunjukkan kelas sentimen negatif, dan nilai polaritas bernilai 0 masuk kedalam kelas netral. Hasil pelabelan dengan TextBlob dapat dilihat pada Gambar 4.12.

	A	B	C	D
1	text	english	polarity	score
2	nikah muda nikah mudah pikir buset pikir nik aja	Marriage Young Marriage Easy Thinking Desem Thinking Nik just th	0,23333333	positif
3	valid mbk embah kakung ibukku tinggal wafat er	Valid mbk grandfather my mother lives my died grandparent of a	0,1375	positif
4	perempuan muda cantik didik kerja kaya hidup e	beautiful young women students work rich lives well marriage pe	0,378571429	positif
5	buka televisi isi suami canda cewek muda cantik	Open the television filled with a husband joking a beautiful young	0,19	positif
6	minimo be like nuna pilih mini baper nih ayo nik	minimo like nuna choose a mini baper, let's get married or not	0,25	positif
7	sih duda muda bawah umur an selang nikah nen	the young widower is underage and the marriage hose is a great -g	0,45	positif
8	terima nasehat teman umur motivator muda dll	accept the advice of young motivator age friends, etc. valid marria	0,1	positif
9	rin muda sakit ya kalo nikah	Young Rin is sick if you get married	-0,121428571	negatif
10	fuck kaget sahur temu teman gendong anak bilai	Fuck was surprised to meet a friend to carry a child saying young n	-0,066666667	negatif
11	as hak paham edukasi rumah tangga edukasi ana	as the right to understand household education children's educati	0,066082251	positif
12	suruh nikah muda pacar aja stres	Tell the young boyfriend's marriage stress	0,1	positif
13	elaee gaya bicara nikah muda gilir dekat cowok a	Elaee Speaking Style Young Marriage Turns Shill Near Guys I'm Laz	0,0525	positif
14	bismillah april moga kerja rumahsakit moga nika	Bismillah April, I hope that the hospital work is to marry young ma	0,1	positif
15	nikah muda biar gapunya anak muda	young marriage so that the gap is young people	0,1	positif
16	mati muda biar gak nikah muda	Dead young so as not to marry young	0	netral
17	besok nikah muda brarti jodoh	Tomorrow is a young marriage	0,1	positif
18	si nikah wajah awet muda ya oh suami tampan k	The marriage face is young, oh oh handsome husband wehehe hel	0,3	positif
19	wkwkwk joint research kampus kampus dosen n	Wkwkwk Joint Research Campus Campus Young Lecturer Cute Mar	0,3	positif
20	nyeri muncul ibu bilang muda habis nikah hmmm	Pain appears Mother says young after marriage hmmm	0,1	positif
21	bayang umur gendong bayi tuduh nikah muda fu	shadow age of carrying baby accusing young marriage fuck	-0,15	negatif
22	masalah duda rumit banding pasang nikah muda	complicated widowing problems.	-0,5	negatif

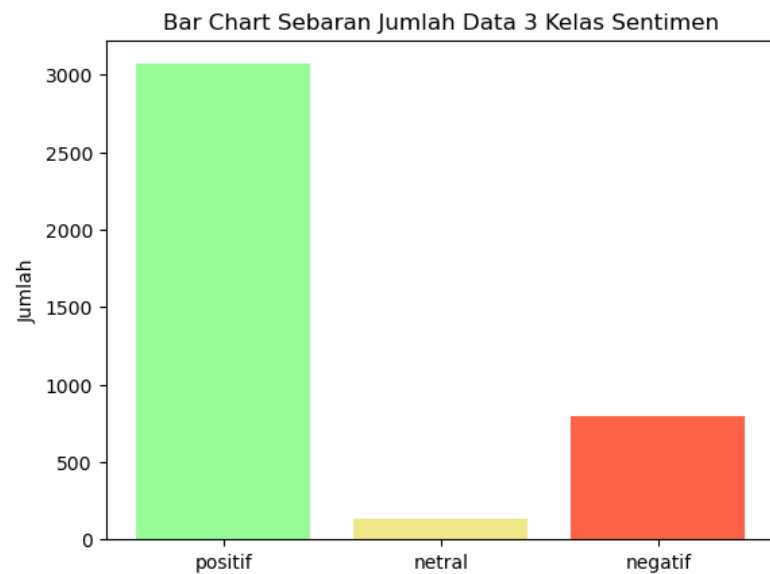
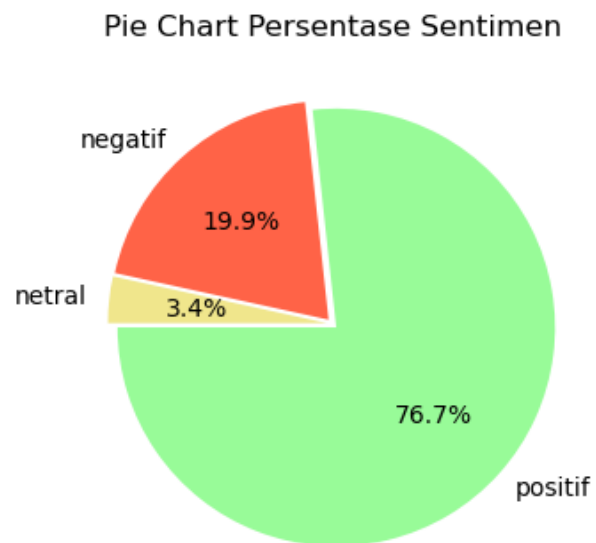
Gambar 4.12 Hasil Pelabelan dengan TextBlob

Didapatkan hasil akhir dari pelabelan dengan menggunakan library textblob sebanyak 4000 data tweet adalah 3069 tweet yang masuk dalam kelas positif, 137 tweet kelas netral, dan 794 tweet kelas negatif yang ditunjukkan di Tabel 4.8.

Tabel 4.8 Tabel Hasil Pelabelan dengan TextBlob

Positif	Netral	Negatif	Jumlah
3069	137	794	4000

Jumlah sebaran data hasil labeling data dengan TextBlob dapat dilihat pada gambar 4.13 dan 4.14.

**Gambar 4.13 Bar Chart Labeling****Gambar 4.14 Pie Chart Labeling**

4.1.4 Ekstraksi Fitur

Hal pertama yang dilakukan dalam tahap ini adalah membagi data menjadi dua bagian yaitu data latih dan data uji. Kode di Gambar 4.15 menghasilkan data uji atau *test size* sebanyak 20% dari total keseluruhan data, dan sisanya yaitu 80% data menjadi data latih. Sehingga kode tersebut menghasilkan output 3200 data latih dan 800 data uji.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, stratify =
y, random_state=1, test_size=0.2, shuffle=True)
```

Gambar 4.15 Kode *Split* Data Latih dan Data Uji

Tahap selanjutnya adalah mengubah isi dataset ke dalam representasi *vector* sekaligus menerapkan *N-Gram*. Pada penelitian ini, digunakan library *scikitlearn* untuk mengambil *CountVectorizer* yang dapat mengubah fitur teks menjadi sebuah representasi *vector*, kemudian parameter *N-Gram* akan menyusun kata yang menjadi *bag of words* berbentuk *unigram*. *Unigram* artinya setiap representasi *vector* akan mewakili 1 kata. Kode *CountVectorizer* bisa dilihat di Gambar 4.16.

```
vectorizer = CountVectorizer(analyzer = 'word', ngram_range=(1,1),
binary=True, stop_words='english')

vectorizer.fit (list(x_train) + list(x_test))

x_train_vec = vectorizer.transform(x_train)
x_test_vec = vectorizer.transform(x_test)
```

Gambar 4.16 Kode *CountVectorizer*

4.1.5 Implementasi SVM

Tahap ini adalah tahap membuat model *Support Vector Machine classifier*.

Pada data latih terdapat kelas sentiment positif, netral, dan negatif. SVM akan mempelajari karakteristik kata-kata yang terdapat pada masing-masing kelas pada data latih tersebut, kemudian SVM akan mencoba memprediksi kelas sentiment pada data uji sebanyak 800 data. Pada tahap pemodelan ini, dicoba tiga kernel SVM yaitu *Linear*, *RBF*, dan *Polynomial* dengan nilai *default* pada *C*, *gamma*, dan *degree* untuk mengetahui kernel apa yang paling baik akurasi dalam mengklasifikasikan 4000 data pada penelitian ini. Kode implementasi SVM dapat dilihat di Gambar 4.17 dan Gambar 4.18.

```
#Membuat Classifier SVM Linear
linear1 = svm.SVC(kernel='linear', C=1)
linear1.fit(x_train_vec, y_train).predict(x_test_vec)

#Membuat Classifier SVM RBF
rbf = svm.SVC(kernel='rbf', gamma='scale', C=1)
rbf.fit(x_train_vec, y_train).predict(x_test_vec)

#Membuat Classifier SVM Polynomial
poly = svm.SVC(kernel='poly', degree=3, C=1)
poly.fit(x_train_vec, y_train).predict(x_test_vec)

linear_pred = linear.predict(x_test_vec)
rbf_pred = rbf.predict(x_test_vec)
poly_pred = poly.predict(x_test_vec)
```

Gambar 4.17 Kode Pelatihan dan Pengujian SVM

```
score_linear = accuracy_score(linear_pred, y_test)
score_rbf = accuracy_score(rbf_pred, y_test)
score_poly = accuracy_score(poly_pred, y_test)

print("Accuracy with Linear SVM: ", score_linear * 100, '%')
print("Accuracy with RBF SVM: ", score_rbf * 100, '%')
print("Accuracy with Poly SVM: ", score_poly * 100, '%')
```

Gambar 4.18 Kode Akurasi SVM

Kode pada Gambar 4.18 menghasilkan output:

Tabel 4.9 Tabel Perbandingan Akurasi Tiga Kernel SVM

No	Kernel SVM	Akurasi
1	Linear	87,375%
2	Radial Basis Function	83,875%
3	Polynomial	78,625%

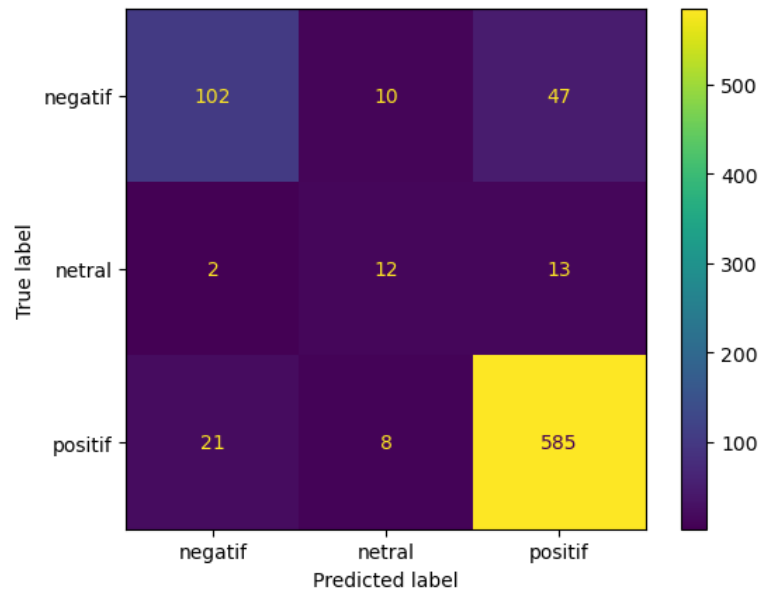
Pada Tabel 4.9 dapat disimpulkan bahwa *Linear SVM* memiliki performa paling baik dalam menguji data *tweet* pada penelitian ini. Maka dari itu, kernel tersebut adalah kernel yang akan digunakan di penelitian ini untuk evaluasi performansi dan sekaligus menjadi *classifier* untuk aplikasi web analisis sentimen.

4.1.6 Evaluasi Performansi

Untuk mengetahui performa dari metode *Linear SVM*, maka dilakukan pengujian terhadap model yang telah dibuat. Hasil klasifikasi akan ditampilkan dalam bentuk *confusion matrix*. *Confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya. Kode bisa dilihat di Gambar 4.19 dan visualisasi di Gambar 4.20.

```
cm = confusion_matrix(y_test, linear_pred, labels=linear.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
                              display_labels=linear.classes_)
disp.plot()
plt.show()
```

Gambar 4.19 Kode Confusion Matrix



Gambar 4.20 Visualisasi *Confusion Matrix*

Hasil evaluasi model *confusion matrix* dapat dilihat di Tabel 4.10.

Tabel 4.10 Tabel *Confusion Matrix*

	Prediksi Kelas		
	Negatif	Netral	Positif
Negatif	102	10	47
Netral	2	12	13
Positif	21	8	585

Model mengklasifikasikan 102 data negatif yang teridentifikasi dengan benar bersentimen negatif, 10 data negatif teridentifikasi netral, dan 47 data negatif teridentifikasi positif. 12 data netral benar teridentifikasi dengan benar bersentimen netral, 2 data netral teridentifikasi negatif, dan 13 data netral teridentifikasi positif. Kemudian 585 data positif teridentifikasi dengan benar bersentimen positif, 21 data positif teridentifikasi negatif, dan 8 data positif teridentifikasi netral.

Setelah mendapatkan hasil *confusion matrix*, selanjutnya dapat dilakukan perhitungan manual nilai akurasi dari model *Linear SVM*.

$$Accuracy = \frac{True\ Positive + True\ Netral + True\ Negative}{Total\ Data\ Uji} \times 100\%$$

$$= \frac{699}{800} \times 100\%$$

$$= 87,375\%$$

Accuracy menggambarkan seberapa besar tingkat akurat model yang telah dibuat dapat mengklasifikasi data dengan benar. *Accuracy* didapatkan dari perhitungan rasio data benar dengan keseluruhan data. Dengan mengetahui besarnya nilai akurasi pada kinerja model *machine learning*, dapat diketahui tingkat kemampuan model dalam mencari ketepatan antara informasi yang diinginkan dengan jawaban yang diberikan oleh model. Tingkat keberhasilan sistem dalam menemukan sebuah informasi dalam penelitian ini sebesar 87,375 %.

Selanjutnya, untuk melihat nilai performa klasifikasi dari setiap kelas dapat diketahui melalui nilai *precision*, *recall*, dan *f1-score*. Potongan kode di Gambar 4.21 digunakan untuk melihat *classification report*.

```
print(classification_report(y_test, linear_pred))
```

Gambar 4.21 Kode *Classification Report*

Gambar 4.22 merupakan tampilan *classification report*.

	precision	recall	f1-score	support
negatif	0.82	0.64	0.72	159
netral	0.40	0.44	0.42	27
positif	0.91	0.95	0.93	614
accuracy			0.87	800
macro avg	0.71	0.68	0.69	800
weighted avg	0.87	0.87	0.87	800

Gambar 4.22 Classification Report

Berdasarkan *classification report* pada gambar 4.22, akan dijelaskan penjabaran perhitungan *precision*, *recall*, dan *f1-score*. Berikut perhitungannya:

1. Perhitungan *Precision*

$$Precision = \frac{TP}{TP+FP}$$

$$Precision \text{ negatif} = \frac{102}{102+23} \times 100\% = 82\%$$

$$Precision \text{ netral} = \frac{12}{12+18} \times 100\% = 40\%$$

$$Precision \text{ positif} = \frac{585}{585+60} \times 100\% = 91\%$$

2. Perhitungan *Recall*

$$Recall = \frac{TP}{TP+FN}$$

$$Recall \text{ negatif} = \frac{102}{102+57} \times 100\% = 64\%$$

$$Recall \text{ netral} = \frac{12}{12+15} \times 100\% = 44\%$$

$$Recall \text{ positif} = \frac{585}{585+29} \times 100\% = 95\%$$

3. Perhitungan *F1-Score*

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

$$F1-Score \text{ negatif} = \frac{2*(0,82*0,64)}{0,82+0,64} 100\% = 72\%$$

$$F1\text{-Score netral} = \frac{2*(0,40*0,44)}{0,40+0,44} 100\% = 42\%$$

$$F1\text{-Score positif} = \frac{2*(0,91*0,95)}{0,91+0,95} 100\% = 93\%$$

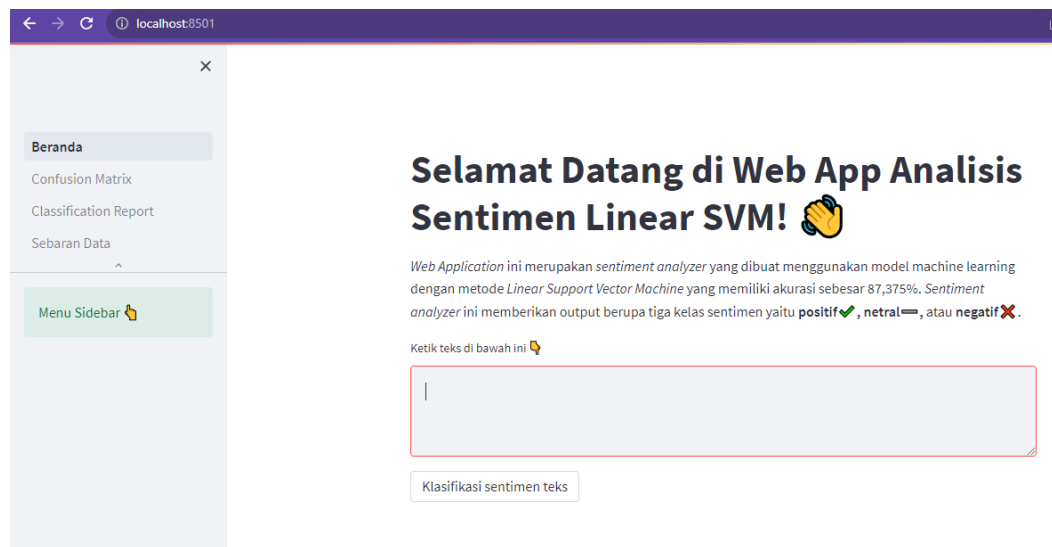
Nilai *precision* untuk kelas negatif sebesar 82%, untuk kelas netral sebesar 40%, untuk kelas positif sebesar 91%. Angka ini dapat diartikan bahwa proporsi presisi label yang diprediksi dengan benar dari total prediksi cukup tinggi untuk kelas positif dan negatif. Sedangkan *recall* untuk kelas negatif sebesar 64%, untuk kelas netral sebesar 44%, dan kelas positif sebesar 95%. Hal ini berarti keberhasilan kinerja sistem dalam menemukan kembali informasi yang bernilai positif dalam dokumen lebih baik dibandingkan dengan menemukan informasi kembali yang bernilai netral dan negatif. Sementara *F1-Score* bisa diartikan sebagai *harmonic mean* (rata-rata yang dihitung dengan cara mengubah semua data menjadi pecahan) dari *precision* dan *recall*. *F1-Score* yang baik mengindikasikan bahwa model klasifikasi memiliki *precision* dan *recall* yang baik.

4.2 Uji Coba dan Pembahasan Sistem

Tahap ini berisi pengujian kinerja sistem beserta pembahasannya

4.2.1 Antarmuka Aplikasi Web

Aplikasi web yang dibuat di penelitian ini memiliki 4 halaman yaitu Beranda, Confusion Matrix, Classification Report, dan Sebaran Data seperti yang bisa dilihat di Gambar 4.23.



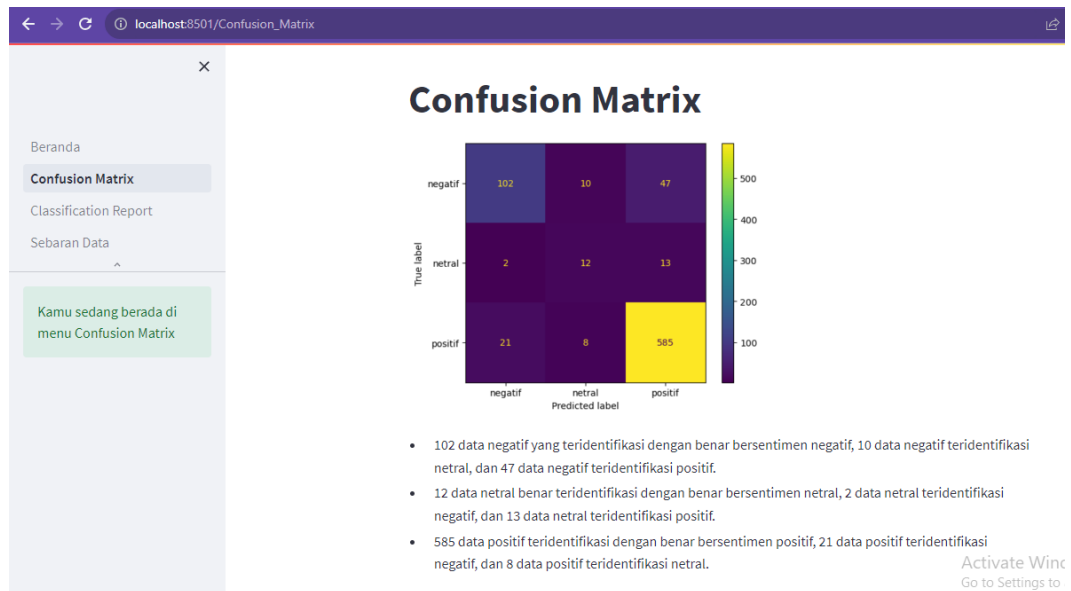
Gambar 4.23 Halaman Beranda

Pada menu Beranda, pengguna akan disambut dengan sebuah *text field* yang berfungsi untuk memasukkan teks untuk menguji kemampuan model SVM dalam mengklasifikasi teks ke dalam salah satu dari tiga kelas sentimen.

Aplikasi web ini merupakan sebuah sistem *sentiment analyzer* yang bisa mendeteksi sentimen dari teks Bahasa Indonesia yang dimasukkan oleh pengguna. Di dalam *back end* aplikasi web tersebut, sudah dimasukkan sebuah model machine learning *Linear SVM* beserta pustaka ekstraksi fitur yaitu *CountVectorizer* dan *N-Gram* yang dilatih dan diuji menggunakan data tweet yang menghasilkan akurasi sebesar 87,375%.

Ketika pengguna menekan tombol “Klasifikasi sentimen teks”, maka teks Bahasa Indonesia yang sudah dimasukkan tersebut akan langsung diterjemahkan oleh sistem ke dalam bahasa Inggris menggunakan *library* *googletrans*. Teks tersebut harus diterjemahkan terlebih dahulu sebelum diklasifikasi oleh model karena model dibuat menggunakan dataset berbahasa Inggris. Lalu, model akan

memasukkan teks yang sudah diterjemahkan tersebut ke dalam 3 kelas sentimen yaitu positif, netral, atau negatif kemudian hasilnya akan ditampilkan.



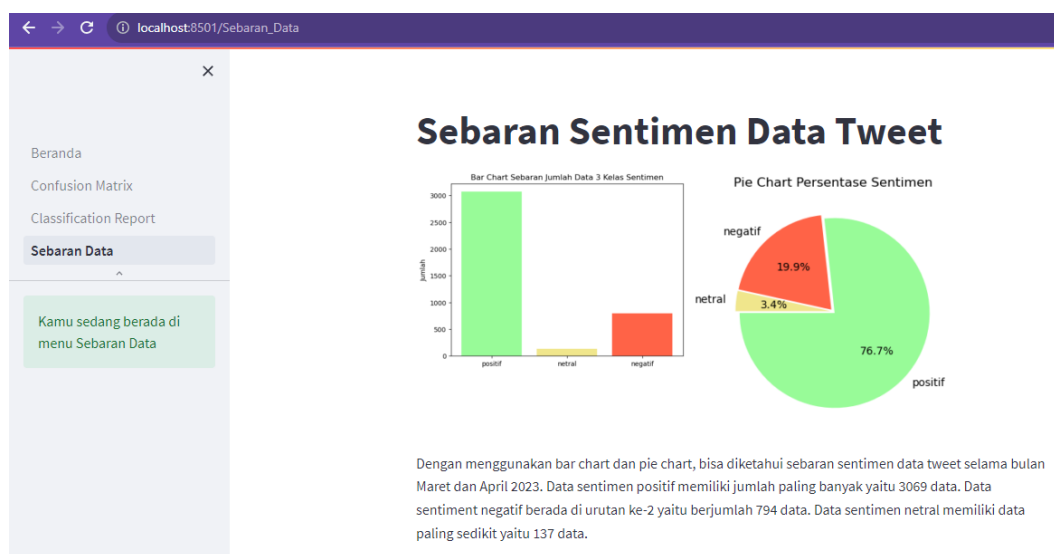
Gambar 4.24 Halaman Confusion Matrix

Kemudian pada menu Confusion Matrix di Gambar 4.24, disediakan tampilan *confusion matrix* dari model Linear SVM yang sudah melalui proses pengujian.



Gambar 4.25 Halaman Classification Report

Kemudian menu selanjutnya adalah “Classification Report” yang bisa dilihat di Gambar 4.25. Menu tersebut memuat hasil perhitungan evaluasi performansi yang digunakan untuk menampilkan nilai *accuracy*, *precision*, *recall*, dan *F1-Score* pada masing-masing kelas sentimen.



Gambar 4.26 Halaman Sebaran Data

Kemudian halaman terakhir bernama “Sebaran Data” yang bisa dilihat di Gambar 4.26 adalah halaman yang memuat infografis tentang penyebaran sentimen data *tweet* pada dataset yang berisi 4000 data *tweet* selama bulan Maret dan April 2023. Data divisualisasikan menggunakan *Bar Chart* dan *Pie Chart*.

4.2.2 Hasil Uji Coba

Pada tahap ini, sistem menjalani uji coba untuk mengklasifikasikan teks yang dimasukkan pengguna ke dalam *text field*. Pertama, sistem akan mencoba untuk mendeteksi kalimat bersentimen negatif seperti yang ada pada Gambar 4.27.

Selamat Datang di Web App Analisis Sentimen Linear SVM! 🤝

Web Application ini merupakan *sentiment analyzer* yang dibuat menggunakan model machine learning dengan metode *Linear Support Vector Machine* yang memiliki akurasi sebesar 87,375%. *Sentiment analyzer* ini memberikan output berupa tiga kelas sentimen yaitu **positif**✅, **netral**➡️, atau **negatif**❌.

Ketik teks di bawah ini 📝

takut nikah muda karena kondisi ekonomi saya sedang buruk

Klasifikasi sentimen teks

['negatif']

Gambar 4.27 Input Teks dengan Hasil Klasifikasi Negatif

Sistem berhasil mengklasifikasikan teks tersebut ke dalam kelas “negatif” untuk kalimat “takut nikah muda karena ekonomi saya sedang buruk”. Kemudian uji coba selanjutnya adalah memasukkan kalimat bernada netral. Kalimat bersentimen netral adalah kalimat yang tidak mengandung unsur pro atau kontra terhadap suatu topik. Uji teks netral ditunjukkan di Gambar 4.28.

Selamat Datang di Web App Analisis Sentimen Linear SVM! 🤝

Web Application ini merupakan *sentiment analyzer* yang dibuat menggunakan model machine learning dengan metode *Linear Support Vector Machine* yang memiliki akurasi sebesar 87,375%. *Sentiment analyzer* ini memberikan output berupa tiga kelas sentimen yaitu **positif**✅, **netral**➡️, atau **negatif**❌.

Ketik teks di bawah ini 📝

biaya pernikahan pasangan itu sekitar tujuh ratus juta rupiah

Klasifikasi sentimen teks

['netral']

Gambar 4.28 Input Teks dengan Hasil Klasifikasi Netral

Pada Gambar 4.28, sistem berhasil mengkalsifikasikan kalimat netral ke dalam kelas netral. Selanjutnya sistem diuji dengan teks bernada positif. Menurut hasil pengujian, model SVM dalam penelitian ini memiliki performa paling tinggi dalam mendeteksi data bersentimen positif. Kali ini sistem diuji dengan kalimat “saya suka dengan gagasan nikah muda” yang bisa dilihat di Gambar 4.29.

Selamat Datang di Web App Analisis Sentimen Linear SVM! 🤝

Web Application ini merupakan *sentiment analyzer* yang dibuat menggunakan model machine learning dengan metode *Linear Support Vector Machine* yang memiliki akurasi sebesar 87,375%. *Sentiment analyzer* ini memberikan output berupa tiga kelas sentimen yaitu **positif**✅, **netral**➡️, atau **negatif**❌.

Ketik teks di bawah ini 🖋️

saya suka dengan gagasan nikah muda

Klasifikasi sentimen teks

['positif']

Gambar 4.29 Input Teks dengan Hasil Klasifikasi Positif

Dengan aplikasi web *sentiment analyzer* ini, model SVM Linear dapat mengklasifikasikan ketiga teks ke dalam tiga sentimen yang sesuai. Berikut rinciannya:

- takut nikah muda karena ekonomi saya sedang buruk = **negatif**
- biaya nikah pasangan itu sekitar tujuh ratus juta rupiah = **netral**
- saya suka dengan gagasan nikah muda = **positif**

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan antara lain:

1. Penelitian ini memberikan hasil analisis sentimen pada topik “nikah muda” di media sosial Twitter menggunakan metode *Support Vector Machine* yang mampu mengklasifikasikan *tweet* ke dalam kelas sentimen positif, netral, dan negatif.
2. Dihasilkan sebuah aplikasi web yang memiliki model machine learning *Linear SVM* di dalam *backend* yang dapat mengklasifikasikan teks yang dimasukkan oleh pengguna ke dalam kelas positif, netral, atau negatif.
3. Hasil akurasi pengujian klasifikasi dengan *Linear Support Vector Machine* adalah 87,375%.
4. Sentimen netral berjumlah sedikit bisa dikarenakan *TextBlob* yang mendeteksi kata “muda” menjadi positif.
5. Sistem memiliki performa paling baik dalam mendeteksi teks bersentimen positif.

5.2 Saran

Hasil penelitian ini masih memiliki banyak kekurangan. Maka dari itu, penelitian ini masih bisa dikembangkan dengan cara-cara sebagai berikut:

1. Menggunakan metode *machine learning* lain terutama metode yang didesain untuk klasifikasi *multiclass* sehingga bisa digunakan sebagai perbandingan hasil uji model untuk mencari metode klasifikasi terbaik.
2. Menggunakan dataset yang jumlah data di setiap kelasnya mendekatiimbang.
3. Melakukan *preprocessing* dengan lebih baik terutama di tahap filtering dan normalisasi.
4. Menggunakan *tool* pelabelan lain seperti VADER (*Valence Aware Dictionary and sEntiment Reasoner*), BERT (*Bidirectional Encoder Representations from Transformers*), atau SpaCy.
5. Menggunakan media sosial lain sebagai media analisis sentimen seperti Reddit, Instagram, Facebook, dan lain-lain.

DAFTAR PUSTAKA

- Anjani, S. A. dan Achmad Fauzan. (2021). Implementasi *n-Gram* dalam Analisis Sentimen Masyarakat DIY Terhadap PSBB Jawa-Bali Jilid II Menggunakan *Naive Bayes Classifier*. *Statistika*, 21(2), 73-83.
- Budiman, I., M. Reza Faisal. dan Dodon T. N. (2018). Studi Ekstraksi Fitur Berbasis Vektor Word2Vec pada Pembentukan Fitur Berdimensi Rendah. *Jurnal Komputasi*, 8(1), 62-69. DOI: <http://dx.doi.org/10.23960%2Fkomputasi.v8i1.2517>.
- Data Indonesia. 2022. Pengguna Twitter di Indonesia Capai 18,45 Juta pada 2022. <https://dataindonesia.id/digital/detail/pengguna-twitter-di-indonesia-capai-1845-juta-pada-2022>. Diakses pada tanggal 13 September 2022 pukul 10.29.
- Ellina, dkk. (2022). Prediksi Keberhasilan Lamaran Pekerjaan Dengan *Count Vectorizer* dan *Logistic Regression*. *Prosiding Seminar Nasional Riset Dan Information Science (SENARIS) 2022*, vol. 4, 16-25. DOI: <http://dx.doi.org/10.30645/senaris.v4i2.204>.
- Ependi, U. dan Ade Putra. (2019). Solusi Prediksi Persediaan Barang dengan Menggunakan Algoritma Apriori (Studi Kasus: Regional Part Depo Auto 2000 Palembang). *Jurnal Edukasi dan Penelitian Informatika*, 5(2), 139-145. DOI: <http://dx.doi.org/10.26418/jp.v5i2.32648>.
- Haranto, F. F. dan Bety Wulan Sari. (2019). Implementasi Support Vector Machine untuk Analisis Sentimen Pengguna Twitter Terhadap Pelayanan Telkom dan Biznet. *Jurnal PILAR Nusa Mandiri*, 15(2), 171-176. DOI: <https://doi.org/10.33480/pilar.v15i2.699>.
- Harruma, Issha. (2022). Kasus Pernikahan Dini di Indonesia. <https://nasional.kompas.com/read/2022/10/02/000000061/kasus-pernikahan-dini-di-indonesia>. Diakses pada tanggal 22 Desember 2022 pukul 21.33.
- Hidayat, W., Mursyid A., dan Arif Setyanto. (2021). Pengaruh Algoritma ADASYN dan SMOTE terhadap Performa Support Vector Machine pada Ketidakseimbangan Dataset Airbnb. *Edumatic: Jurnal Pendidikan Informatika*, 5(1), 11-20. DOI: <https://doi.org/10.29408/edumatic.v5i1.3125>.
- Himawan, R. D. dan Eliyani. (2022). Perbandingan Akurasi Analisis Sentimen Tweet terhadap Pemerintah Provinsi DKI Jakarta di Masa Pandemi. *JEPIN:*

Jurnal Edukasi dan Penelitian Informatika, 7(1), 58-63. DOI: <http://dx.doi.org/10.26418/jp.v7i1.41728>.

Husada, H. C. dan Adi S. P. Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM). *TEKNIKA: Jurnal Teknologi dan Informasi*, 10(1), 18-26. DOI: 10.34148/teknika.v10i1.311.

Indrayuni, E. (2019). Klasifikasi *Text Mining Review* Produk Kosmetik Untuk Teks Bahasa Indonesia Menggunakan *Algoritma Naive Bayes*. *Jurnal Khatulistiwa Informatika*, 7(1), 29-36. DOI: <https://doi.org/10.31294/jki.v7i1.5740.g3245>.

JustAnotherArchivist. (2023). Snsrape. <https://github.com/JustAnotherArchivist/snsrape>. Diakses pada tanggal 28 Mei 2023 pukul 13.24.

Karsito dan Santi Susanti. (2019). Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah dengan Algoritma *Naive Bayes* di Perumahan Azzura Residencia. *Jurnal Teknologi Pelita Bangsa*, 9(3), 43-48.

Mubaroroh H. H., Hasbi Y., dan Agus R. (2022). Analisis Sentimen Data Ulasan Aplikasi Ruangguru pada Situs Google Play menggunakan Algoritma *Naive Bayes Classifier* dengan Normalisasi Kata *Levenshtein Distance*. *Jurnal Gaussian*, 11(2), 248-257. DOI: <https://doi.org/10.14710/j.gauss.v11i2.35472>.

Munawar dan Yosua Riadi Silitonga. (2019). Sistem Pendeteksi Berita Hoax di Media Sosial dengan Teknik Data Mining Scikit Learn. *Jurnal Ilmu Komputer*, 4(2), 173-179. DOI: <https://doi.org/10.47007/komp.v4i02.3140>.

Nugroho, Agung. (2018). Analisis Sentimen Pada Media Sosial Twitter Menggunakan *Naive Bayes Classifier* Dengan Ekstraksi Fitur *N-Gram*. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 2(2), 200-209. DOI: <http://dx.doi.org/10.30645/j-sakti.v2i2.83>.

Paramastri, N. A. dan Gumgum Gumilar. (2019). Penggunaan Twitter Sebagai Medium Distribusi Berita dan Newsgathering oleh Tirto.id. *Kajian Jurnalisme*, 3(1), 18-38. DOI : <https://doi.org/10.24198/jkj.v3i1.22450>.

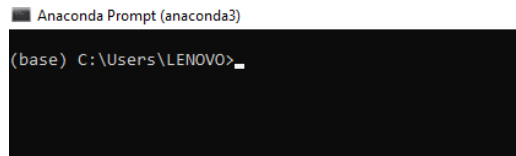
Parapat, I. M., Muhammad Tanzil F., dan Sutrisno. (2018). Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(10), 3163-3169.

- Pravina, A. M., Imam Cholissodin, dan Putra Pandu Adikara. (2019). Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(3), 2789-2797.
- Prihatini, Putu Manik. (2017). Implementasi Ekstraksi Fitur pada Pengolahan Dokumen Berbahasa Indonesia. *Jurnal Matrix*, 6(3), 174-178.
- Python Software Foundation. (2022). What is Python? Executive Summary. <https://www.python.org/doc/essays/blurb/>. Diakses pada tanggal 15 November 2022 pukul 10.47.
- Python Software Foundation. (2022). Whetting Your Appetite. <https://docs.python.org/3/tutorial/appetite.html>. Diakses pada tanggal 15 November 2022 pukul 15.27.
- Rahutomo, F., Pramana Y. S., dan Miftahul A. F. (2018). Implementasi Twitter Sentiment Analysis Untuk Review Film Menggunakan Algoritma *Support Vector Machine*. *Jurnal Informatika Polinema*, 4(2), 93-100. DOI: 10.33795/jip.v4i2.152.
- Republik Indonesia. (2019). Undang – Undang No. 16 Tahun 2019 tentang perubahan atas Undang – Undang Nomor 1 tahun 1974 tentang Perkawinan. Lembaran Negara RI Tahun 2019, No. 186. Sekretariat Negara. Jakarta.
- Retnoningsih, E. dan Rully Pramudita. (2020). Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python. *Bina Insani ICT Journal*, 7(2), 156-165. DOI: <https://doi.org/10.51211/biict.v7i2.1422>.
- Roihan, A., Po Abas Sunarya, dan Ageng Setiani Rafika. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *Indonesian Journal on Computer and Information Technology*, 3(1), 75-82. DOI: <https://doi.org/10.31294/ijcit.v5i1.7951>.
- Septian, J. A., Tresna Maulana Fahrudin, dan Aryo Nugroho. (2019). Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor. *INSYST: Journal of Intelligent System and Computation*, 1(1), 43-49. DOI: <https://doi.org/10.52985/insyst.v1i1.36>.
- Serliana. (2020). Dampak Positif dan Negatif Pernikahan Dini Yang Perlu Diketahui. <https://ringtimesbali.pikiran-rakyat.com/kesehatan/pr-28645727/dampak-positif-dan-negatif-pernikahan-dini-yang-perlu-diketahui>. Diakses pada tanggal 12 September 2022 pukul 08.52.

- Simorangkir, H. dan Kemas Muslim Lhaksamana. (2018). Analisis Sentimen pada Twitter untuk Games Online Mobile Legends dan Arena of Valor dengan Metode Naïve Bayes Classifier. *e-Proceeding of Engineering*, 5(3), 8131-8140.
- Suanpang P., Pitchaya J., dan Phuripoj K. (2021). Sentiment Analysis with a TextBlob Package Implications for Tourism. *Journal of Management Information and Decision Sciences*, 24(6), 1-9.
- Suyanto. (2017). *Data Mining untuk Klasifikasi dan Klasterisasi Data*. Bandung: Informatika Bandung.
- Twitter Inc. (2022). What is Twitter?. <https://help.twitter.com/en/resources/new-user-faq>. Diakses pada tanggal 9 November 2022 pukul 20.21.
- Tuhuteru, H. dan Ade Iriani. (2018). Analisis Sentimen Perusahaan Listrik Negara Cabang Ambon Menggunakan Metode Support Vector Machine dan Naive Bayes Classifier. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 3(3), 394-401. DOI: <http://dx.doi.org/10.30591/jpit.v3i3.977>.
- Zalyhaty, L. Q., Vivine N., dan Erwin S. (2020). Analisis Sentimen Tanggapan Masyarakat Terhadap Vaksin Covid-19 Menggunakan Algoritma Support Vector Machine (SVM). *JSIKA: Jurnal Sistem informasi Universitas Dinamika*, 9(4), 1-10.

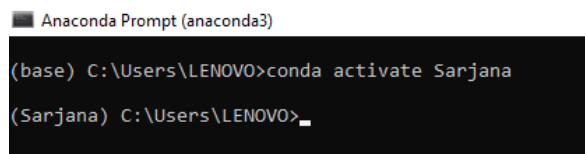
CARA MENJALANKAN PROGRAM

1. Buka Anaconda Prompt



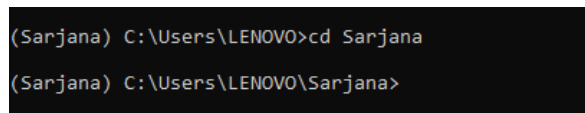
```
Anaconda Prompt (anaconda3)
(base) C:\Users\LENOVO>
```

2. Aktifkan *environment* tempat menampung *libraries* dan pekerjaan untuk penelitian ini.



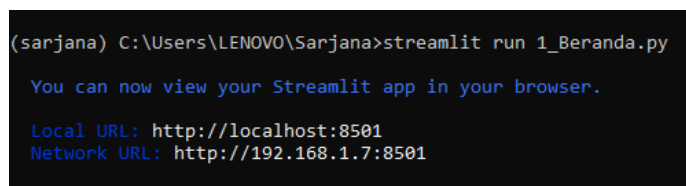
```
Anaconda Prompt (anaconda3)
(base) C:\Users\LENOVO>conda activate Sarjana
(Sarjana) C:\Users\LENOVO>
```

3. Setelah *environment* aktif, masuk ke direktori tempat file aplikasi web berada.



```
(Sarjana) C:\Users\LENOVO>cd Sarjana
(Sarjana) C:\Users\LENOVO\Sarjana>
```

4. Jalankan aplikasi web.



```
(sarjana) C:\Users\LENOVO\Sarjana>streamlit run 1_Beranda.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.1.7:8501
```

5. Browser otomatis terbuka dan aplikasi sudah berjalan

LISTING PROGRAM

1. Import *Libraries*

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import snsrape.modules.twitter as sntwitter
import openpyxl
import string
import re #regex library
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter
import nltk
from textblob import TextBlob
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

2. Proses *Scraping*

```
query = "nikah muda lang:id since:2023-03-01 until:2023-04-01"
tweets = []

for i, tweet in enumerate(sntwitter.TwitterSearchScraper(query).get_items()):
    if i >= 2000:
        break
    else:
        tweets.append([tweet.date, tweet.user.username, tweet.content])

maret = pd.DataFrame(tweets, columns = ['date', 'username', 'tweet'])
maret['date'] = maret['date'].dt.tz_localize(None)
maret.to_excel('maret.xlsx', index=False)
```

```

query = "nikah muda lang:id since:2023-04-01 until:2023-05-01"
tweets = []

for i, tweet in
enumerate(sntwitter.TwitterSearchScrapper(query).get_items
()):
    if i >= 2000:
        break
    else:
        tweets.append([tweet.date, tweet.user.username,
tweet.content])

april = pd.DataFrame(tweets, columns =
['date', 'username', 'tweet'])
april['date'] = april['date'].dt.tz_localize(None)
april.to_excel('april.xlsx', index=False)

```

3. Proses *Preprocessing*

```

data = pd.read_excel('data_latih.xlsx')
data
df=pd.DataFrame(data[['tweet']])
df

# ===== CASE FOLDING =====
df['case folding'] = df['tweet'].str.lower()
df

# ===== DATA CLEANSING =====
def remove_tweet_special(text):
    # remove tab, new line, and back slice
    text = text.replace('\t', " ").replace('\n', "
").replace('\u', " ").replace('\ ', "")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii',
'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9
]+)|(\w+:\/\/\/\S+)", " ", text).split())
    # remove incomplete URL
    return text.replace("http://", "
").replace("https://", " ")

df['data cleansing'] = df['case folding'].apply
(remove_tweet_special)
df.head()

```

```

#remove number
def remove_number(text):
    return re.sub(r"\d+", "", text)

df['data cleansing'] = df['data
cleansing'].apply(remove_number)

#remove punctuation
def remove_punctuation(text):
    return
text.translate(str.maketrans("", "", string.punctuation))

df['data cleansing'] = df['data
cleansing'].apply(remove_punctuation)

#remove whitespace leading & trailing
def remove_whitespace_LT(text):
    return text.strip()

df['data cleansing'] = df['data
cleansing'].apply(remove_whitespace_LT)

#remove multiple whitespace into single whitespace
def remove_whitespace_multiple(text):
    return re.sub('\s+', ' ', text)

df['data cleansing'] = df['data
cleansing'].apply(remove_whitespace_multiple)

# remove single char
def remove_single_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", text)

df['data cleansing'] = df['data
cleansing'].apply(remove_single_char)
df.head()

# ==== TOKENIZING ====
def tokenization(text):
    return word_tokenize(text)

df['tokenization'] = df['data
cleansing'].apply(tokenization)
df

# ==== NORMALISASI ====

normalized_word = pd.read_excel("normalisasi.xlsx")
normalized_word_dict = {}

```



```

for index, row in normalized_word.iterrows():
    if row[0] not in normalized_word_dict:
        normalized_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalized_word_dict[term] if term in
normalized_word_dict else term for term in document]

df['normalisasi'] =
df['tokenization'].apply(normalized_term)
df

# ==== FILTERING ====

# get stopwords indonesia
list_stopwords = stopwords.words('indonesian')

# ----- manually add stopwords -----
# append additional stopwords
list_stopwords.extend(['nik','ais','ih','kuea','ndes','tk',
',','arghhhh','wuakakak','gtth','wowww','apeeee','Aksjsjsk',
',','alae','koq','wuakakak','salengpraew','rukhhadevata','g',
'tth','zeon','vivienne','yaam','woyy','ykwim','auff','ue',
',','hoek','hayo','chnmn','hahahah','haaaaaa','din','woy','nd',
'eer','lalalala','wkwkwwkwkwkw','woyyy','dih','den','heheh',
'ew','etdah','beeeuh','wahh','heheee','hhaaha','waaaaa','o',
'akilah','haaaahh','huft','ai','et','acha','ue','hokyahoky',
'a','hahahihi','yl','wihh','hahahaa','hhhh','def','ayom','',
'ser','duh','heuheuheu','huwaaaaa','yalah','mww','cekabi',
'a','dikatar','angganara','krtsk','woee','ndi','ohh','www',
',','aee','huaaaa','gn','hahahah','nd','ema','ceratops','pa',
'suk','ygy','repp','gais','hadehhhhhhh','walah','hahah','p',
'aa','awkwkwk','wkwkk','wkkwk','wkwkwwkwkwkwah','wkwkwwk',
',','baceprot','sksksk','heheh','brooo','dbd','aeee','weeeh',
',','wehh','milta','hsnah','swsg','hemm','xda','yara','ohh','h',
'eh','kle','acy','hayooo','hahahahaha','balablablabla','la',
'i','loj','itine','heehehe','kwkwk','kwkwkwkwkwkwk','waaa',
',','demending','pali','eeh','dlsb','cooooy','hehehehe','adje',
'm','aih','syar','wkwkk','aowkwkwk','walah','euy','der','h',
'ahaa','hesteg','hmmmmmtar','gtideologi','ab','owkwkwkwk',
',','dncw','sloga','jo','jengjenggg','anuanu','caw','ehheheheh',
',','hlaa','hahahihi','ckckckck','sich','pakin','mmarkpkk',
',','ponponpon','kyary','pamyu','laaahhh','cp','duhhh','napen',
',','lise','bi','ieu','poho','boga','imah','keur','ulin','k',
'wkkwk','ehheh','gryli','oalah','prekk','hehh','cere','eke',
'kekek','chco','nganu','wkwkwwkwkwkwkwk','pfft','awowkwkwkw',
'k','kinyis','pus','yng','yg','yang','kwoswkw','wkwkwwkw',
'kwk','ahahha','weeeeh','hah','fir','hong','jay','haikyuu',
',','nderrrr','omtanteuwaksodara','ahsajkakaka','kwkwkwk','d',
'errr','wwkwkwkw','hadehh','aaaaa','heeh','dem','ocaaa','w'

```

```

o','prenup','dihhh','cokk','imho','chenle','jsdieksisnisa
wikwok','hahahahahahaha','bam','yowohh','lau','boiiiiii','
gih','beuhhh','wkw','wkwkwkw','dooong','oalaaaa','sinoeng
','wkekwk','nyai','cai','anw','tjuyyy','hanss','mh','ih',
'widihh','cy','eeeeee','gi','luat','laaaaa','cam','lancau'
,'tuch','kun','uhhhh','chuakssss','oiyaa','hadeuhhhh','wk
wkwkwkw','hehehee','nk','lak','qwq','oneesan','eeehmmm','
am','wkwk'])

#---- add stopwords from txt file ----

txt_stopword = pd.read_csv("stopwordbahasa.txt", names=
["stopwords"], header = None)

list_stopwords.extend(txt_stopword["stopwords"][0].split(
' '))
list_stopwords = set(list_stopwords)

def stopwords_removal(words):
    return [word for word in words if word not in
list_stopwords]

df['stopwords'] =
df['normalisasi'].apply(stopwords_removal)
df

# ==== STEMMING ====
# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmer
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in df['stopwords']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("-----")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":", term_dict[term])

print(term_dict)
print("-----")

```

```

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df['stemmed'] =
df['stopwords'].swifter.apply(get_stemmed_term)
print(df['stemmed'])

==== penggabungan kata ====
def fit_stopwords(text):
    text= np.array(text)
    text= ' '.join(text)
    return text

df['text']=df['stemmed'].apply(lambda x:
fit_stopwords(x))
df
df.to_excel("bahasaindonesia.xlsx", index=False)

```

4. Proses *Labeling*

```

==== LABELING dengan TEXTBLOB ====
data = pd.read_excel('english.xlsx')
data

df=pd.DataFrame(data[['text','english']])
df

for tweet in df.english:
    clean_tweet= tweet

    blob_object = TextBlob(clean_tweet)
    hasil= blob_object.tags

    print(hasil)

polarity = lambda x: TextBlob(x).sentiment.polarity
df['polarity']=df['english'].apply(polarity)
df

def analysis(score):
    if score > 0:
        return 'positif'
    elif score == 0:
        return 'netral'
    else:
        return 'negatif'

df['score']=df['polarity'].apply(analysis)

```

```

df

print("positif :", (sum(df['score']=='positif')))
print("netral :", (sum(df['score']=='netral')))
print("negatif :", (sum(df['score']=='negatif')))

#Membuat BarChart
x_axis = ['positif', 'netral', 'negatif']
y_axis = [(sum(df['score']=='positif')),
(sum(df['score']=='netral')),
(sum(df['score']=='negatif'))]
c = ['palegreen', 'khaki', 'tomato']
plt.bar(x_axis, y_axis, color = c)
plt.title('Bar Chart Sebaran Jumlah Data 3 Kelas Sentimen')
plt.ylabel('Jumlah')
plt.show()

#Membuat PieChart
plt.figure(figsize=(8,4))
plt.title("Pie Chart Persentase Sentimen", fontsize=12)
chart =
plt.pie(df.score.value_counts(),explode=(0.025,0.025,0.025),
        labels=df.score.value_counts().index,
        colors=['palegreen','tomato','khaki'],
        autopct='%1.1f%%', startangle=180)
plt.show()

df.to_excel("labeled.xlsx")

```

5. Proses Ekstraksi Fitur

```

==== EKSTRAKSI FITUR ====
data = pd.read_excel('labeled.xlsx')
data

df=pd.DataFrame(data)

y=df.score.values
x=df.english.values

x_train, x_test, y_train, y_test = train_test_split(x, y,
stratify = y, random_state=1,
                                                    test_size=0.2,
shuffle=True)

print(x_train.shape)
print(x_test.shape)

```

```

vectorizer = CountVectorizer(analyzer = 'word',
ngram_range=(1,1), binary=True, stop_words='english')
vectorizer.fit (list(x_train) + list(x_test))

x_train_vec = vectorizer.transform(x_train)
x_test_vec = vectorizer.transform(x_test)
print(x_train_vec.shape)
print(x_test_vec.shape)

```

6. Proses Klasifikasi Support Vector Machine

```

==== KLASIFIKASI SVM ====
#Membuat Classifier SVM Linear
linear = svm.SVC(kernel='linear', C=1)
linear.fit(x_train_vec, y_train).predict(x_test_vec)

#Membuat Classifier SVM RBF
rbf = svm.SVC(kernel='rbf', gamma='scale', C=1)
rbf.fit(x_train_vec, y_train).predict(x_test_vec)

#Membuat Classifier SVM Polynomial
poly = svm.SVC(kernel='poly', degree=3, C=1)
poly.fit(x_train_vec, y_train).predict(x_test_vec)

linear_pred = linear.predict(x_test_vec)
rbf_pred = rbf.predict(x_test_vec)
poly_pred = poly.predict(x_test_vec)

score_linear = accuracy_score(linear_pred, y_test)
score_rbf = accuracy_score(rbf_pred, y_test)
score_poly = accuracy_score(poly_pred, y_test)

print("Accuracy with Linear SVM: ",score_linear * 100,
'%.1f')
print("Accuracy with RBF SVM: ",score_rbf * 100, '%.1f')
print("Accuracy with Polynomial SVM: ",score_poly * 100,
'%.1f')

cm = confusion_matrix(y_test, linear_pred,
labels=linear.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=linear.classes_)

disp.plot()
plt.show()

print("Linear SVM Confusion Matrix: ")
print(confusion_matrix(y_test, linear_pred))

print(classification_report(y_test, linear_pred))

```

7. Halaman Beranda Aplikasi Web

```
import streamlit as st
import streamlit as st
from googletrans import Translator
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import openpyxl
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import
CountVectorizer
from sklearn import svm

st.write("# Selamat Datang di Web App Analisis Sentimen
Linear SVM! 🤖")

st.markdown(
    """
        *Web Application* ini merupakan *sentiment analyzer*
yang dibuat menggunakan model
        machine learning dengan metode *Linear Support Vector
Machine* yang
        memiliki akurasi sebesar 87,375%. *Sentiment
analyzer* ini memberikan output
        berupa tiga kelas sentimen yaitu **positif✅,
netral➡️,** atau **negatif❌**.
    """
)

st.sidebar.success("Menu Sidebar 📁")

#-----#

#Read data
data = pd.read_excel('labeled.xlsx')
df=pd.DataFrame(data)

#Split data training dan testing
y=df.score.values
x=df.english.values
x_train, x_test, y_train, y_test = train_test_split(x, y,
stratify = y, random_state=1,
                                                    test_size=0.2,
shuffle=True)

#CountVectorizer dan N-Gram
vectorizer = CountVectorizer(analyzer = 'word',
ngram_range=(1,1), binary=True, stop_words='english')
vectorizer.fit (list(x_train) + list(x_test))
```

```

x_train_vec = vectorizer.transform(x_train)
x_test_vec = vectorizer.transform(x_test)

#Membuat Classifier SVM Linear
linear = svm.SVC(kernel='linear', C=1)
linear.fit(x_train_vec, y_train).predict(x_test_vec)

#Input teks
translator = Translator()
message = st.text_area("Ketik teks di bawah ini 📌")
try:
    translated_text =
str(translator.translate(message,src='id',dest='en'))
except TypeError:
    pass

#Klasifikasi
if st.button("Klasifikasi sentimen teks"):
    text_vector = vectorizer.transform([translated_text])
    st.success(linear.predict(text_vector))

```

8. Halaman *Confusion Matrix* Aplikasi Web

```

import streamlit as st
from PIL import Image

st.sidebar.success("Kamu sedang berada di menu Confusion
Matrix")

st.title("Confusion Matrix")
image = Image.open('confusionmatrix.png')
st.image(image, width = 400)
st.markdown(
    """
- 102 data negatif yang teridentifikasi dengan
    benar bersentimen negatif, 10 data negatif
    teridentifikasi netral,
    dan 47 data negatif teridentifikasi positif.
- 12 data netral benar teridentifikasi dengan benar
    bersentimen netral, 2 data netral teridentifikasi
    negatif, dan 13 data netral teridentifikasi
    positif.
- 585 data positif teridentifikasi dengan benar
    bersentimen positif, 21 data positif
    teridentifikasi
    negatif, dan 8 data positif teridentifikasi netral.
    """
)

```

9. Halaman *Classification Report* Aplikasi Web

```
import streamlit as st
from PIL import Image

st.sidebar.success("Kamu sedang berada di menu
Classification Report")

st.title("Classification Report")
image = Image.open('classreport.png')
st.image(image, width = 400)
st.markdown(
    """
        - Nilai precision untuk kelas negatif sebesar 82%,
          kelas netral sebesar 40%, kelas positif sebesar
          91%.
          Artinya presisi proporsi label yang diprediksi
          dengan benar dari
            total prediksi cukup tinggi untuk kelas positif dan
            negatif.
        - Nilai recall kelas negatif sebesar 64%, kelas
          netral sebesar 44%, dan
          kelas positif sebesar 95%. Artinya kinerja
          keberhasilan sistem
            dalam menemukan kembali informasi yang bernilai
            positif dalam dokumen
            lebih baik dibandingkan dengan menemukan informasi
            kembali yang
            bernilai netral dan negatif.
        - Nilai F1-Score untuk kelas negatif sebesar 72%,
          kelas netral sebesar 42%, dan kelas positif sebesar
          93%. F1-Score bisa
            diartikan sebagai harmonic mean (rata-rata yang
            dihitung dengan cara
            mengubah semua data menjadi pecahan) dari precision
            dan recall.
            F1-Score yang baik mengindikasikan bahwa model
            klasifikasi memiliki
            precision dan recall yang baik.
    """
)
```

10. Halaman *Classification Report* Aplikasi Web

```
import streamlit as st
from PIL import Image


st.sidebar.success("Kamu sedang berada di menu Sebaran
Data")
```




```
st.title("Sebaran Sentimen Data Tweet")
images = ['bar.png', 'pie.png']
st.image(images, width=150 * len(images))
st.markdown(
    """
    Dengan menggunakan bar chart dan pie chart, bisa
    diketahui sebaran
    sentimen data tweet selama bulan Maret dan April
    2023. Data sentimen positif
    memiliki jumlah paling banyak yaitu 3069 data. Data
    sentiment negatif
    berada di urutan ke-2 yaitu berjumlah 794 data. Data
    sentimen netral
    memiliki data paling sedikit yaitu 137 data.

    """
)
```

KRITERIA, CATATAN, DAN KEPUTUSAN PENDADARAN

1	PEMBERITAHUAN SEBELUM UJIAN :Pengumpulan akhir dokumen Tugas Akhir/Skripsi melewati batas akhir ganjil 2022/2023, mahasiswa harus menyelesaikan registrasi dan KRS semester berikutnya.			
2				
3	KRITERIA KELULUSAN UJIAN SIDANG / PENDADARAN			
4				
5				
6	1. Lulus ujian tanpa syarat, disebut kriteria 1.			
7	2. Lulus bersyarat, disebut kriteria 2, yaitu dengan sedikit perbaikan atau penyempurnaan text dan atau program dal 30 September 2023			
8	dan tidak ada ujian lagi. Jika dalam waktu yang ditentukan mahasiswa tersebut tidak dapat menyelesaikan, maka, mahasiswa yang bersangkutan dianggap tidak lulus ujian.			
9	3. Tidak lulus ujian sidang/pendadaran, disebut kriteria 3, dijelaskan, disarankan Ketua Tim Penguji untuk			
10				
11	Ketentuan bagi peserta yang tidak lulus ujian sidang / pendadaran.			
12	1) Mahasiswa wajib menempuh ujian sidang/pendadaran ulang			
13	2) Kesempatan ujian sidang/pendadaran ulang hanya diberikan dalam rentang waktu maksimum 6 bulan, setelah			
14	3) Jika sampai batas waktu maksimum 6 bulan tersebut belum dapat diajukan/diselesaikan, maka calon peserta			
15	4) Mahasiswa yang akan menempuh ujian sidang/pendadaran ulang ini diwajibkan membayar biaya ujian setara 2			
16				
17				
18				
19	Yogyakarta, 21 Juli 2023			
20	Memahami dan			
21	Mematuhi peraturan di			
22				
23				
24				
25				
26	Nama Mahasiswa			
27	RADEN ISNAWAN ARGY ARYASATYA			

		YAYASAN PENDIDIKAN WIDYA BAKTI YOGYAKARTA UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA Jl. Raya Janti (Majapahit) No.143, Yogyakarta, 55198, Telp (0274) 486664, Website: www.utdi.ac.id , E-mail: info@utdi.ac.id			
Hari, tanggal : Jum'at, 21 Juli 2023 Waktu : 10.00 Nama : RADEN ISNAWAN ARGY ARYASATYA No. Mahasiswa / Juru : 195410257 / Informatika					
No		Hal yang harus diperbaiki		Pemberi Catatan	
1.		kebutuhan output diganti menjadi hasil analisis. bukan aplikasi web. cek lagi undang2 perkawinan (cek yg terbaru).		B. Sari	
2.		Tambah dataset sebanyak 2000 data. Kata "suka" tidak dikenali oleh sistem. alasan memilih Kernel Linear karena blm ada di naskah(masukan ke ruang lingkup dan di bab 3). hasil precision, recall ditambahkan di interface		B. Ariesta	
3.		datanya perbanyak yg negatif dan netral			

		YAYASAN PENDIDIKAN WIDYA BAKTI YOGYAKARTA UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA Jl. Raya Janti (Majapahit) No.143, Yogyakarta, 55198, Telp (0274) 486664, Website: www.utdi.ac.id , E-mail: info@utdi.ac.id			
KEPUTUSAN HASIL UJIAN PENDADARAN					
Sesuai dengan hasil sidang pendadaran pada tanggal					
maka					
Nama Mahasiswa RADEN ISNAWAN ARGY ARYASATYA					
NIM / Program Studi 195410257 / Informatika					
Jenjang S1					
dinyatakan LULUS dengan kriteria LULUS					
Ketua Penguji Sari Iswanti, S.Si., M.Kom.					

ACC REVISI

Acc revisi
Misti
Anista D.
3/8/2023

Acc revisi
Handy
Maria Mediatra
1/8/2023

acc revisi
Kedi
1/8/2023

SKRIPSI

ANALISIS SENTIMEN TWITTER TERHADAP PERNIKAHAN DI USIA
MUDA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE
(SVM)



RADEN ISNAWAN ARGY ARYASATYA

NIM: 195410257

PROGRAM STUDI INFORMATIKA

PROGRAM SARJANA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

YOGYAKARTA

2023

SURAT KETERANGAN
PERSETUJUAN PUBLIKASI

Bahwa yang bertanda tangan di bawah ini:

Nama : Raden Isnawan Argi Aryasatya
NIM : 195410257
Jurusan : Informatika
Email : zargi.teddy7@gmail.com
Judul Skripsi : Analisis Sentimen Twitter Terhadap Pernikahan Di Usia Muda
Menggunakan Metode Support Vector Machine (SVM)

Menyerahkan karya ilmiah kepada pihak perpustakaan UTDI dan menyetujui untuk **diunggah ke Repository** Perpustakaan UTDI sesuai dengan ketentuan yang berlaku untuk kepentingan riset dan pendidikan.

Yogyakarta, 3 Agustus 2023

Penulis,



Raden Isnawan Argi Aryasatya

NIM : 195410257

