

**ANALISIS SENTIMEN APLIKASI
INVESTASI REKSA DANA (BIBIT)
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE***

TUGAS AKHIR

Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana Informatika
Universitas Pembangunan Nasional “Veteran” Yogyakarta



Disusun oleh :

**ANDRE SAPUTRA
123160194**

**PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
YOGYAKARTA
2023**

HALAMAN PENGESAHAN PEMBIMBING

ANALISIS SENTIMEN APLIKASI INVESTASI REKSA DANA (BIBIT) MENGUNAKAN METODE *SUPPORT VECTOR MACHINE*

Disusun Oleh :

Andre Saputa
123160194

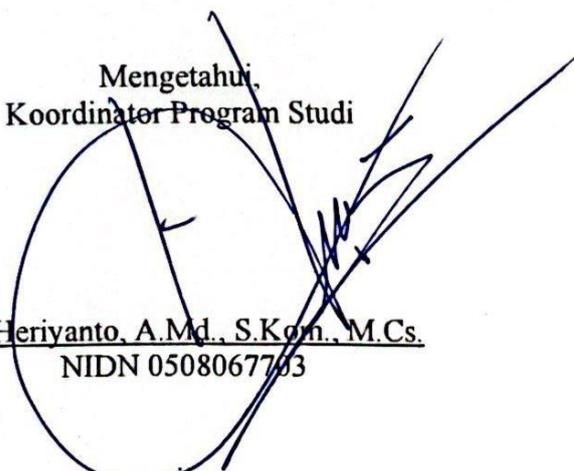
Telah diuji dan dinyatakan **Lulus** oleh pembimbing
pada tanggal : 12 Juli 2023

Menyetujui,
Pembimbing



Budi Santosa, S.Si., M.T.
NIDN 0510097001

Mengetahui,
Koordinator Program Studi



Dr. Heriyanto, A.Md., S.Kom., M.Cs.
NIDN 0508067793

HALAMAN PENGESAHAN PENGUJI

ANALISIS SENTIMEN APLIKASI INVESTASI REKSA DANA (BIBIT) MENGUNAKAN METODE *SUPPORT VECTOR MACHINE*

Disusun Oleh :

Andre Saputa

123160194

Telah diuji dan dinyatakan **Lulus** oleh penguji
pada tanggal : 12 Juli 2023

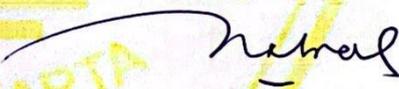
Menyetujui,

Penguji I

Penguji II


Budi Santosa, S.Si., M.T.

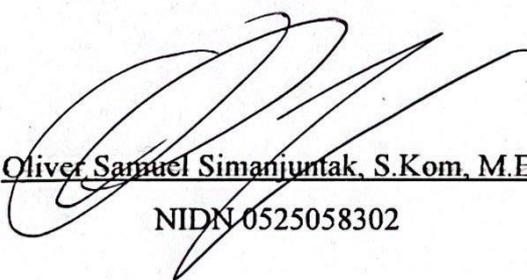
NIDN 0510097001


Nur Heri Cahyana, S.T., M.Kom.

NIDN 022096003

Penguji III

Penguji IV


Oliver Samuel Simanjuntak, S.Kom, M.Eng.

NIDN 0525058302


Rifki Indra Perwira, S.Kom, M.Eng.

NIDN 0508078301

SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertanda tangan di bawah ini, saya:

Nama : Andre Saputra

NIM : 123160194

Menyatakan bahwa karya ilmiah saya yang berjudul:

**ANALISIS SENTIMEN APLIKASI INVESTASI REKSA DANA (BIBIT)
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE***

Merupakan karya asli saya dan belum pernah dipublikasikan dimanapun. Apabila ada di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apapun yang diberikan Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta

Pada Tanggal :

Yang menyatakan,



(Andre Saputra)

PERNYATAAN BEBAS PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama : Andre Saputra

NIM : 123160194

Fakultas/Prodi : Teknik Industri/Informatika

Dengan ini saya menyatakan bahwa judul tugas akhir:

**ANALISIS SENTIMEN APLIKASI INVESTASI REKSA DANA (BIBIT)
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE***

Adalah hasil kerja saya sendiri dan benar bebas dari plagiat kecuali cuplikan serta ringkasan yang terdapat di dalamnya setelah saya jelaskan sumbernya (sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No. 17 Tahun 2000 dan pernyataan dan perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Dibuat di: Yogyakarta

Pada Tanggal:

Yang Menyatakan,



(Andre Saputra)

HALAMAN PERSEMBAHAN

Alhamdulillahirabbil alamin

“Dengan segala puji syukur ke hadirat Allah SWT yang telah melimpahkan karunia-Nya, sehingga saya bisa menyelesaikan tugas akhir ini.

*Tugas akhir ini saya persembahkan kepada orang-orang tercinta ayah, ibu, adik.
Terima kasih atas segala dukungan dan doanya.”*

“Justru karena dunia ini kejam, kita jadi sadar dan menikmati bahwa hal-hal kecil yang membuat bahagia adalah hal-hal yang berharga” -Fiony Alveria Tantri

ABSTRAK

Pengguna aktif aplikasi Bibit tercatat sebanyak 85 ribu orang, angka tersebut menunjukkan bahwa Bibit merupakan aplikasi investasi reksa dana terbanyak yang digunakan oleh orang Indonesian dalam berinvestasi reksa dana. Pendapat yang diberikan pengguna aplikasi Bibit pun beragam, mulai dari yang suka terhadap aplikasi Bibit, maupun yang tidak suka terhadap aplikasi Bibit. Akan tetapi nilai bintang yang diberikan oleh pengguna aplikasi Bibit pada halaman ulasan terkadang tidak cocok dengan komentar yang diberikan, sehingga ulasan tersebut tidak terklasifikasikan dengan akurat. Padahal jika komentar yang diberikan dapat diketahui apakah bersifat positif, atau negatif maka dapat membantu untuk menjadi salah satu tolak ukur untuk mengetahui bagus atau tidaknya aplikasi bibit untuk berinvestasi reksa dana. Untuk itu maka diperlukan suatu teknik pengklasifikasian komentar yang disebut analisis sentimen.

Penelitian akan dilakukan untuk mengklasifikasikan jenis ulasan dari aplikasi investasi reksa dana Bibit dalam Bahasa Indonesia ke dalam kategori sentimen positif dan negatif. Penelitian ini menggunakan data yang diperoleh dari halaman *web* aplikasi Bibit pada *google play store* yang diambil menggunakan teknik *web scraping*, di mana data tersebut diberi label positif dan negatif. Kemudian data akan dilakukan *pre-processing*, pembobotan setiap kata atau fitur menggunakan *term frequency - inverse document frequency* (TF-IDF) dan selanjutnya dilakukan proses *training* menggunakan kernel *Radial Basis Function* (RBF) pada metode *Support Vector Machine* (SVM). Pengujian dilakukan menggunakan data sebanyak 2000 data dengan perbandingan data latih sebanyak 70% dan data uji sebanyak 30%.

Hasil pengujian yang dilakukan menunjukkan bahwa kernel *Radial Basis Function* (RBF) menghasilkan nilai akurasi sebesar 90%, nilai rata-rata *recall* sebesar 90%, dan nilai rata-rata presisi sebesar 90.1%.

Kata Kunci : Analisis Sentimen, RBF, SVM, Reksa Dana

ABSTRACT

There are 85 thousand active users of Bibit application, this figure shows that Bibit is the largest number of mutual fund investment applications used by Indonesians to invest in mutual funds. The reviews given by the users of Bibit application also varied, ranging from those who liked Bibit to those who did not like Bibit. However, the star rating given by Bibit application users on the review page sometimes does not match the comments given, so these reviews are not classified accurately. In fact, if the comments given can be seen whether they are positive or negative, then they can help to become one of the benchmarks to find out whether or not Bibit for investing in mutual funds is good. For this reason, we need a comment classification technique called sentiment analysis.

Research will be conducted to classify the types of reviews of the mutual fund investment application Bibit in Indonesian into positive and negative sentiment categories. This study used data obtained from Bibit application web page on the Google Play Store which was taken using a web scraping technique, where the data was labeled positive and negative. Then the data will be pre-processed, weighing each word or feature using the term frequency - inverse document frequency (TF-IDF) and then the training process will be carried out using the Radial Basis Function (RBF) kernel on the Support Vector Machine (SVM) method. The test was carried out using 2000 data with a comparison of 70% training data and 30% test data.

The test results showed that the Radial Basis Function (RBF) kernel produced an accuracy value of 90%, an average recall value of 90%, and an average precision value of 90.1%.

Keywords : *Sentiment Analysis, RBF, SVM, Mutual fund*

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT, yang telah memberikan petunjuk dan kekuatan dalam menyelesaikan Tugas Akhir yang berjudul “Analisis Sentimen Aplikasi Investasi Reksa Dana (Bibit) Menggunakan Metode *Support Vector Machine*”. Tugas akhir ini merupakan syarat terakhir yang harus ditempuh untuk menyelesaikan pendidikan pada jenjang Strata Satu (S1) Program Studi Informatika Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Dalam penyusunan tugas akhir ini, penulis menyadari bahwa penulisan ini tidak lepas dari bimbingan, bantuan serta dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT yang senantiasa memberikan kesehatan, petunjuk dan kemudahan selama menyelesaikan tugas akhir.
2. Ayah dan Ibu yang telah sabar untuk selalu mendukung dan mendoakan penulis dalam menyelesaikan tugas akhir ini.
3. Bapak Budi Santosa, S.Si., M.T., selaku dosen pembimbing atas waktu, pengertian, segala bantuan dalam memberikan referensi dan inovasi, pemberian kritik dan saran yang membangun, dan juga pemberian semangat kepada penulis.
4. Seluruh Dosen dan Pegawai jurusan Informatika atas kebaikan, dukungan, dan bantuan yang diberikan kepada penulis dalam menyelesaikan tugas akhir.
5. Semua rekan yang telah membantu penyusunan tugas akhir.
6. Para perempuan hebat di kelompok sanggar tari FX Sudirman.

Penulis menyadari penyusunan tugas akhir ini tentunya tidak lepas dari kekurangan dan masih jauh dari sempurna. Oleh karena itu penulis mengharapkan masukan berupa kritik, saran, dan tanggapan yang bersifat membangun dalam upaya pembelajaran lebih lanjut.

Akhir kata semoga tugas akhir ini bermanfaat bagi semua pihak, dan penulis sendiri pada khususnya.

Yogyakarta, Juli 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN PEMBIMBING	ii
HALAMAN PENGESAHAN PENGUJI	iii
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR	iv
PERNYATAAN BEBAS PLAGIAT.....	v
HALAMAN PERSEMBAHAN.....	vi
ABSTRAK	vii
ABSTRACT	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah	4
1.3. Batasan Masalah	4
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian dan Pengembangan Sistem.....	4
1.6.1. Metodologi Penelitian	5
1.6.2. Metodologi Pengembangan Sistem.....	6
1.7. Sistematika Penulisan	7
BAB II TINJAUAN LITERATUR	9
2.1. Reksa Dana	9
2.2. Bibit.....	9
2.3. Analisis Sentimen	9
2.4. Klasifikasi Teks	10
2.5. <i>Web Scraping</i>	11
2.6. <i>Text Pre-processing</i>	12
2.6.1. <i>Cleansing</i>	12

2.6.2.	<i>Word Normalization</i>	12
2.6.3.	<i>Tokenizing</i>	12
2.6.4.	<i>Casefolding</i>	13
2.6.5.	<i>Stopword Removal</i>	13
2.6.6.	<i>Stemming</i>	14
2.7.	<i>Term Frequency-Inverse Document Frecuency (TF-IDF)</i>	14
2.8.	<i>Support Vector Machine</i>	14
2.9.	<i>Confusion Matrix</i>	17
2.10.	<i>Studi Pustaka</i>	18

BAB III METODE PENELITIAN DAN PENGEMBANGAN SISTEM..... 20

3.1.	Metode Penelitian	20
3.1.1.	Analisis Masalah	21
3.1.2.	Pengumpulan Data	21
3.1.2.1.	Studi literatur	21
3.1.2.2.	<i>Web scraping</i>	21
3.1.3.	Pelabelan Data	23
3.1.4.	<i>Text Pre-processing</i>	24
3.1.4.1.	<i>Flowchart tahap cleansing</i>	24
3.1.4.2.	<i>Flowchart tahap word conversion</i>	25
3.1.4.3.	<i>Flowchart tahap tokenizing</i>	27
3.1.4.4.	<i>Flowchart tahap casefolding</i>	28
3.1.4.5.	<i>Flowchart tahap stopwords removal</i>	29
3.1.4.6.	<i>Flowchart tahap stemming</i>	30
3.1.5.	Pembobotan Kata TF-IDF.....	31
3.1.5.1.	Perhitungan TF.....	32
3.1.5.2.	Perhitungan DF, IDF dan W	33
3.1.6.	Tahap Pelatihan SVM	34
3.1.6.1.	Tentukan bidang pembatas.....	35
3.1.6.2.	Ubah format data latih ke dalam format <i>support vector</i>	35
3.1.6.3.	Hitung kernelisasi	35
3.1.7.	Tahap Pengujian SVM.....	36

3.1.8.	Pengujian.....	37
3.2.	Metode Pengembangan Sistem.....	38
3.2.1.	Pengumpulan Kebutuhan.....	38
3.2.1.1.	Pengumpulan data.....	38
3.2.1.2.	Analisis kebutuhan sistem.....	39
3.2.2.	Perancangan.....	40
3.2.2.1.	Perancangan arsitektur.....	40
3.2.2.2.	Perancangan antarmuka.....	40
BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN.....		43
4.1.	Hasil Penelitian.....	43
4.1.1.	Pengumpulan Data.....	43
4.1.2.	Pelabelan Data.....	44
4.1.3.	Implementasi Analisis Sentimen Aplikasi Bibit.....	45
4.1.3.1.	Modul <i>text pre-processing</i>	45
4.1.3.2.	Modul pembobotan kata TF-IDF.....	50
4.1.3.3.	Modul <i>support vector machine</i>	52
4.1.4.	Implementasi Sistem.....	53
4.2.	Pengujian.....	56
4.3.	Pembahasan.....	56
BAB V PENUTUP.....		57
5.1.	Kesimpulan.....	57
5.2.	Saran.....	57
DAFTAR PUSTAKA.....		58

DAFTAR GAMBAR

Gambar 2.1	Ilustrasi Metode <i>Support Vector Machine</i>	15
Gambar 3.1	Metode Penelitian Kuantitatif.....	20
Gambar 3.2	Diagram <i>flowchart web scraping</i>	22
Gambar 3.3	Hasil <i>web scraping</i>	22
Gambar 3.4	Diagram <i>flowchart text pre-processing</i>	24
Gambar 3.5	Diagram <i>flowchart cleansing</i>	25
Gambar 3.6	Diagram <i>flowchart</i> tahap <i>word conversion</i>	26
Gambar 3.7	Diagram <i>flowchart</i> tahap <i>tokenizing</i>	27
Gambar 3.8	Diagram <i>flowchart</i> tahap <i>casefolding</i>	28
Gambar 3.9	Diagram <i>flowchart</i> tahap <i>stopword removal</i>	29
Gambar 3.10	Diagram <i>flowchart</i> tahap <i>stemming</i>	30
Gambar 3.11	Diagram <i>flowchart</i> TF-IDF	31
Gambar 3.12	Diagram <i>flowchart</i> pelatihan SVM.....	34
Gambar 3.13	Diagram <i>flowchart</i> pengujian SVM	37
Gambar 3.14	Diagram metode pengembangan sistem.....	38
Gambar 3.15	Rancangan halaman cek ulasan	41
Gambar 3.16	Rancangan halaman data	41
Gambar 3.17	Rancangan halaman <i>confusion matrix</i>	42
Gambar 4.1	Hasil <i>testing</i> SVM	53
Gambar 4.2	Halaman cek ulasan	53
Gambar 4.3	Halaman data	54
Gambar 4.4	Halaman <i>confusion matrix</i>	55

DAFTAR TABEL

Tabel 2.1	Contoh <i>Cleansing</i>	12
Tabel 2.2	Contoh <i>Word Normalization</i>	12
Tabel 2.3	Contoh <i>Tokenizing</i>	13
Tabel 2.4	Contoh <i>Casefolding</i>	13
Tabel 2.5	Contoh <i>Stopword Removal</i>	13
Tabel 2.6	Contoh <i>Stemming</i>	14
Tabel 2.7	<i>Confusion Matrix</i>	17
Tabel 2.8	Tabel Studi Pustaka	18
Tabel 3.1	Contoh hasil pelabelan data	23
Tabel 3.2	Contoh penerapan <i>cleansing</i>	25
Tabel 3.3	Contoh penerapan <i>word conversion</i>	26
Tabel 3.4	Contoh penerapan <i>tokenizing</i>	27
Tabel 3.5	Contoh penerapan <i>casefolding</i>	28
Tabel 3.6	Contoh penerapan <i>stopword removal</i>	30
Tabel 3.7	Contoh penerapan <i>stemming</i>	31
Tabel 3.8	Perhitungan TF	32
Tabel 3.9	Lanjutan perhitungan TF	33
Tabel 3.10	Perhitungan DF, IDF dan TF-IDF	33
Tabel 3.11	Format data <i>support vector</i>	35
Tabel 3.12	Tabel rancangan <i>Confusion Matrix</i>	37
Tabel 3.13	Kebutuhan perangkat keras	39
Tabel 3.14	Kebutuhan perangkat lunak	40
Tabel 4.1	Contoh data hasil <i>web scraping</i>	44
Tabel 4.2	Contoh data hasil pelabelan	44
Tabel 4.3	Lanjutan contoh data hasil pelabelan	45
Tabel 4.4	Contoh data hasil <i>cleansing</i>	45
Tabel 4.5	Lanjutan contoh data hasil <i>cleansing</i>	46
Tabel 4.6	Contoh data hasil <i>word conversion</i>	46
Tabel 4.7	Lanjutan contoh data hasil <i>word conversion</i>	47
Tabel 4.8	Contoh data hasil <i>tokenizing</i>	47
Tabel 4.9	Contoh data hasil <i>casefolding</i>	48
Tabel 4.10	Contoh data hasil <i>stopword removal</i>	49

Tabel 4.11	Contoh data hasil <i>stemming</i>	50
Tabel 4.12	Contoh hasil pembobotan kata TF-IDF	50
Tabel 4.13	Lanjutan contoh hasil pembobotan kata TF-IDF.....	51

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Mengutip dari berita pers yang dikeluarkan oleh PT Kustodian Sentral Efek Indonesia (KSEI) per 3 November 2022, jumlah investor pasar modal yang mengacu pada *Single Investor Identification* (SID) telah mencapai 10.000.628, dengan komposisi jumlah investor lokal sebesar 99,78%. Jumlah tersebut meningkat 33,53% jika dibandingkan dengan akhir tahun 2021 sebesar 7.489.337. Tren peningkatan tersebut telah terlihat sejak tahun 2019 ketika investor masih berjumlah 2 juta. Peningkatan jumlah investor sejak tahun 2019 hingga 2021 merupakan yang tertinggi sepanjang sejarah pasar modal Indonesia. Dari data di atas dapat disimpulkan bahwa tindakan terbaik untuk menjaga keamanan finansial di masa depan adalah berinvestasi.

Menurut buku berjudul *Dasar-dasar Manajemen Investasi* yang ditulis oleh (Nuzula & Nurlaily, 2020) ada berbagai macam pengertian investasi, yaitu investasi sebagai kegiatan mengelola aset berharga. Investasi juga diartikan sebagai kesediaan seseorang (atau investor) mengalokasikan uang dalam nilai tertentu di masa sekarang guna memperoleh penerimaan di kemudian hari. Penerimaan di kemudian hari tersebut dapat dinyatakan sebagai kompensasi yang diterima investor atas komitmennya untuk tidak mengambil uang sebelum masa penerimaan pembayaran di masa datang. Dengan asumsi investor tidak mengambil dananya sebelum periode investasi berakhir, maka besarnya penerimaan tersebut merupakan fungsi dari jangka waktu komitmen investor, tingkat inflasi, serta ketidakpastian atas penerimaan di masa mendatang. Semakin lama periode komitmen, makin besar kompensasi yang diterima investor.

Ada banyak bentuk investasi yang bisa dilakukan, mulai dari investasi yang melibatkan aset nyata atau riil berupa tanah, bangunan, emas, pabrik, mesin dan lain sebagainya. Dan juga investasi berupa aset keuangan atau finansial berupa saham, deposito atau obligasi, atau reksa dana. Dari beberapa bentuk investasi yang tersedia, investasi dalam bentuk aset keuangan atau finansial menjadi salah satu bentuk investasi yang cukup ramah untuk pemula yang ingin mulai berinvestasi. Dan jenis investasi reksa dana yang mendapatkan perhatian lebih, terlebih untuk pemula di mana investasi reksa dana bisa dibeli mulai dari harga terendah Rp10.000,-.

Aset investasi reksa dana pun dapat dengan mudah dibeli oleh calon investor. Ada banyak perusahaan yang menyediakan layanan membeli aset reksa dana melalui perangkat pintar dengan aplikasi-aplikasi seperti Bibit, Bareksa, Ajaib, dan lain sebagainya. Bibit yang hadir sebagai salah satu aplikasi yang menyediakan layanan jual beli reksa dana menjadi salah satu aplikasi paling banyak digunakan oleh orang Indonesia untuk berinvestasi reksa dana. Dengan jumlah pengguna aktif 85 ribu dan telah diunduh sebanyak 5 juta kali menjadi yang paling banyak digunakan dibandingkan aplikasi lainnya yang ada di *google play store*.

Banyaknya jumlah pengguna tersebut belum bisa menjadi penentu kualitas yang diberikan oleh aplikasi Bibit. Salah satu cara yang bisa dilakukan oleh calon investor terlebih yang masih baru dan belum mengerti banyak hal dalam berinvestasi adalah dengan melihat ulasan-ulasan yang sudah diberikan oleh orang-orang yang sudah menggunakan layanan aplikasi Bibit. Karena salah satu faktor yang sering dijadikan pedoman oleh pengguna layanan aplikasi adalah kepuasan pengguna atau konsumen dari layanan aplikasi tersebut. Maka dari itu perlu adanya analisis sentimen pada ulasan-ulasan yang diberikan kepada aplikasi Bibit untuk mengetahui aspek penilaian apa saja yang disampaikan oleh pengguna, sehingga dapat ditentukan klasifikasi sentimen dari hasil ulasan tersebut.

Penelitian analisis sentimen ini bertujuan untuk menguji sentimen orang-orang yang sudah pernah menggunakan aplikasi Bibit dan pada penelitian penulis akan menggunakan algoritma *Support Vector Machine* (SVM) dalam menganalisis sentimen pengguna Bibit yang diambil dari ulasan-ulasan pengguna yang ada di halaman *web google play store*. Pemilihan penelitian dengan menggunakan algoritma *Support Vector Machine* karena SVM mempunyai kelebihan yaitu bisa diterapkan untuk data yang berdimensi tinggi dalam keterbatasan data sampel yang ada (Nugroho et al., 2003), tetapi SVM sulit untuk digunakan untuk data dengan jumlah yang besar dikarenakan proses training yang memakan waktu cukup lama (Chandani & Wahono, 2015).

Pada penelitian analisis sentimen ini penulis akan menggunakan *Radial Basis Function* (RBF) sebagai kernel yang digunakan dengan metode *Support Vector Machine* (SVM). Pemilihan penggunaan kernel RBF dikarenakan menurut (Husada & Paramita, 2021) kernel tersebut menjadi kernel yang sering digunakan dan terbukti efektif dalam melakukan analisis sentimen. Kernel RBF dapat mengatasi pemisahan data yang kompleks dan non-linear dengan lebih baik. Kernel RBF memiliki kemampuan untuk menyesuaikan dengan keberagaman dan interaksi fitur secara lebih fleksibel.

Data ulasan yang akan digunakan dalam penelitian bersumber dari ulasan yang telah diberikan pengguna aplikasi Bibit pada halaman *web google play store*. Data yang telah diambil dari *google play store* akan diproses menjadi data latih dan data uji. Data latih digunakan sebagai bahan belajar agar menghasilkan model yang mampu mengenali pola-pola sentimen, sedangkan data uji digunakan untuk mengukur tingkat akurasi dan performa yang didapatkan dalam proses pengklasifikasian sentimen. Penelitian ini mengklasifikasikan sentimen menjadi 2 parameter, yaitu positif dan negatif.

Pada berbagai penelitian terkait yang pernah dilakukan sebelumnya, dalam melakukan analisis sentimen dapat menggunakan berbagai macam metode seperti *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Naïve Bayes Classifier*. Dari penelitian sebelumnya yang telah dilakukan oleh (Mahendrajaya et al., 2019) tentang analisis sentimen pengguna *gopay* menggunakan metode *Lexicon Based* dan *Support Vector Machine*, dengan hasil 89,17% dengan menggunakan kernel Linear dan 84,38% untuk kernel *Polynomial*. Sedangkan pada penelitian yang dilakukan oleh (Utami, 2017) tentang Analisis Sentimen Opini Publik Berita Kebakaran Hutan dengan menambahkan algoritma *Particle Swarm Optimization* dapat meningkatkan akurasi pada metode *Support Vector Machine* dari yang sebelumnya 80.83% menjadi 87,11%. Dan penelitian (Anggraini & Suroyo, 2019) yang membandingkan analisis sentimen pengguna *digital payment* “*T-cash* dan *gopay*” di sosial media menggunakan *Orange Data Mining* mendapat kesimpulan bahwa *T-cash* memiliki nilai sentimen positif lebih tinggi dibandingkan *gopay* dan memiliki sentimen negatif yang lebih rendah dari pada *gopay*. Namun persamaannya *T-cash* dan *gopay* memiliki kesamaan pola grafik di mana sentimen terbesar adalah netral, diikuti oleh positif, dan terakhir adalah negatif.

Berdasarkan penelitian-penelitian sebelumnya menjadikan dasar dilakukannya penelitian ini, penelitian akan menggunakan algoritma *Support Vector Machine* dengan menggunakan kernel RBF untuk mengetahui tingkat akurasi, rata-rata *recall* dan rata-rata presisi dalam melakukan analisis sentimen pengguna layanan aplikasi Bibit. Sentimen yang diuji akan dibagi menjadi dua kelas, yaitu positif dan negatif. Penelitian ini diharapkan dapat membantu mengambil keputusan orang-orang yang berniat ingin mulai berinvestasi reksa dana dengan aplikasi Bibit.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan di atas, maka dapat diambil rumusan masalah dalam penelitian ini adalah bagaimana tingkat akurasi, rata-rata *recall*, dan rata-rata presisi dari kernel *Radial Basis Function* (RBF) pada algoritma *Support Vector Machine* (SVM) dalam analisis sentimen pada ulasan aplikasi investasi reksa dana Bibit pada *google play store*.

1.3. Batasan Masalah

Pada penelitian ini, permasalahan dibatasi sebagai berikut:

1. Data penelitian didapatkan dari ulasan di halaman *web google play store*.
2. Data penelitian yang digunakan dalam Bahasa Indonesia.
3. Analisis sentimen yang dilakukan hanya menggunakan dua kelas, yaitu positif dan negatif.
4. Dilakukan koreksi terhadap kesalahan penulisan.
5. Penelitian menggunakan analisis sentimen metode *text mining*.

1.4. Tujuan Penelitian

Tujuan pada penelitian ini adalah untuk mengetahui tingkat akurasi, *recall*, dan presisi yang didapatkan dari penggunaan algoritma *Support Vector Machine* dengan kernel RBF dalam analisis sentimen pada aplikasi investasi reksa dana Bibit.

1.5. Manfaat Penelitian

Dengan dilakukannya penelitian ini dapat mengetahui performa dari algoritma *Support Vector Machine* dalam menyelesaikan permasalahan tentang analisis sentimen pada ulasan aplikasi investasi reksa dana Bibit. Selain itu, diharapkan dengan adanya sistem ini dapat mempermudah masyarakat dalam mengambil keputusan untuk mulai berinvestasi reksa dana di aplikasi Bibit.

1.6. Metodologi Penelitian dan Pengembangan Sistem

Adapun metodologi penelitian dan pengembangan sistem yang digunakan pada penelitian ini adalah sebagai berikut:

1.6.1. Metodologi Penelitian

Metodologi penelitian yang akan dilakukan antara lain:

a. Analisis Masalah

Analisis masalah merupakan suatu gambaran masalah yang diangkat dalam penulisan tugas akhir Analisis Sentimen Aplikasi Investasi Reksa Dana Bibit Menggunakan Metode *Support Vector Machine*. Penelitian ini dilakukan untuk merancang sebuah sistem yang dapat memperlihatkan tingkat akurasi dan presisi dari metode *Support Vector Machine* yang digunakan dalam penelitian ini.

b. Studi Literatur

Dalam penelitian ini, studi literatur akan dilaksanakan dengan mencari dan mengumpulkan data dan sumber pustaka yang relevan sehingga dapat mendukung penelitian serta memberikan informasi untuk menyelesaikan permasalahan pada penelitian ini. Studi literatur dapat bersumber dari buku, jurnal, *paper*, maupun artikel yang dapat mendukung pelaksanaan penelitian ini.

c. Pengumpulan Data

Pada bagian ini dilakukan pengumpulan data teks ulasan aplikasi Bibit yang dilakukan dengan proses *web scraping*. *Web scraping* dilakukan untuk mengumpulkan data ulasan aplikasi Bibit dari *google play store*.

d. Pelabelan Data

Data yang telah terkumpul selanjutnya akan dilakukan pelabelan secara manual dengan label Komentar untuk ulasan pengguna Bibit dan Sentimen untuk label positif dan negatif. Proses pelabelan data dilakukan secara manual dengan cara melakukan pengecekan terhadap komentar yang diberikan pengguna aplikasi Bibit. *Score* yang diberikan oleh pengguna pada halaman *web google play store* dijadikan pedoman tambahan dalam melakukan pelabelan data.

e. *Text Pre-processing*

Text pre-processing dilakukan untuk membersihkan data konten yang menjadi masukan dari sistem. Proses ini dilakukan untuk membuat data sama rata dan siap digunakan. *Text pre-processing* memiliki beberapa tahapan yang menjadi sub-proses di antaranya adalah *cleansing*, *word conversion*, *tokenizing*, *casefolding*, *stopword removal* dan *stemming*.

f. Pembobotan Kata

Pembobotan kata merupakan proses yang dilakukan untuk memberikan nilai atau bobot terhadap setiap kata. Pembobotan kata pada penelitian ini menggunakan fitur ekstraksi TF-IDF yang membobotkan kata dengan cara mengalikan hasil *term frequency* (TF) dengan hasil *inverse document frequency* (IDF).

g. Pelatihan *Support Vector Machine*

Pada tahap ini akan menggambarkan proses pelatihan *Support Vector Machine*. Hasil dari pembobotan TF-IDF yang telah dihasilkan sebelumnya akan menjadi data latih pada tahap ini. Tahap ini dilakukan dengan beberapa proses diantaranya adalah mendapat input dari hasil pembobotan kata TF-IDF, menentukan bidang pembatas, mengubah ke dalam format *support vector*, menghitung kernelisasi, menghitung nilai *lagrange multiplier*, menghitung nilai w dan b kemudian akan didapatkan hasil keluaran berupa hasil data latih.

h. Analisis dan Perancangan

Analisis dan perancangan sistem yang akan dibuat yaitu meliputi analisis dari kebutuhan sistem yang akan dirancang. Tahap selanjutnya setelah analisis kebutuhan sistem adalah membuat rancangan sistem mulai dari *flowchart*, dan desain antarmuka pengguna.

i. Pengujian dan Evaluasi

Pada tahap ini akan dilakukan pengujian terhadap sistem untuk mengetahui nilai akurasi dari sistem yang telah dibuat, dan mengevaluasi hasil pengujian tersebut dari *error* dan hasil sesuai dengan kebutuhan yang diharapkan.

1.6.2. Metodologi Pengembangan Sistem

Metodologi penelitian yang akan dilakukan antara lain:

a. Analisis Kebutuhan

Tahap ini merupakan proses mencari informasi mengenai kebutuhan dalam perancangan dan pembangunan sistem.

b. Desain Sistem

Proses untuk membuat rancangan awal sistem atau *blueprint* yang diambil dari hasil analisis kebutuhan yang sudah dilakukan sebelumnya. Desain sistem yang akan dilakukan meliputi perancangan antarmuka atau *interface* dan basis data.

c. Penulisan Kode Program

Tahapan di mana desain sistem yang telah dibuat sebelumnya akan direalisasikan ke dalam sistem dengan mengubah desain sistem menjadi bentuk yang dapat dimengerti oleh komputer, yaitu ke dalam bahasa pemrograman melalui proses *coding*.

d. Pengujian Program

Pengujian program bertujuan untuk memastikan keseluruhan sistem yang telah dibuat dapat berjalan dengan baik dan terbebas dari *error*. Selain itu juga untuk memastikan bahwa keluaran yang dihasilkan oleh sistem sudah sesuai dengan yang diharapkan.

e. Pemeliharaan Program

Tahap pemeliharaan program dilakukan pada sistem yang telah dibuat yang membutuhkan penyesuaian dan atau terdapat kebutuhan baru, namun bukan dimaksudkan untuk pembuatan sistem baru dari awal.

1.7. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam menyusun laporan penelitian ini adalah sebagai berikut:

Bab I Pendahuluan

Pada bagian ini membahas tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Tinjauan pustaka memuat tentang dasar teori yang digunakan untuk analisis dan perancangan sistem serta implementasi pada penelitian ini. Selain itu juga sebagai bahan referensi dan fondasi untuk memperkuat argumentasi dalam penelitian ini. Teori-teori yang sesuai dengan penelitian ini antara lain investasi, reksa dana, Bibit, analisis sentimen, dan *Support Vector Machine*.

Bab III Hasil, Pengujian dan Pembahasan

Pada bab ini akan menyajikan hasil penelitian yang berisi hasil implementasi dari perancangan yang telah dibuat pada bab sebelumnya. Selain itu berisi pengujian terhadap hasil penelitian beserta pembahasannya.

Bab IV Kesimpulan dan Saran

Pada bagian ini berisi kesimpulan dari hasil penelitian dan saran yang diajukan oleh penulis untuk pengembangan penelitian selanjutnya.

BAB II

TINJAUAN LITERATUR

2.1. Reksa Dana

Reksa dana adalah investasi pada aset finansial yang berbentuk wadah dan pola pengelolaan dana/modal bagi sekumpulan pemilik modal untuk berinvestasi dalam instrumen-instrumen investasi yang tersedia di pasar modal dengan cara membeli unit penyertaan reksa dana (Adhianto, 2020). Dana ini kemudian dikelola oleh Manajer Investasi (MI) untuk diinvestasikan ke berbagai investasi, seperti saham, obligasi, pasar uang ataupun efek/sekuritas lainnya.

Menurut Undang-undang Pasar Modal nomor 8 Tahun 1995 pasal 1, ayat (27): “Reksa dana adalah wadah yang dipergunakan untuk menghimpun dana dari masyarakat Pemodal untuk selanjutnya diinvestasikan dalam portofolio Efek oleh Manajer Investasi”.

Masih menurut (Adhianto, 2020) terdapat tiga unsur penting dalam reksa dana yaitu: adanya kumpulan masyarakat baik individu ataupun institusi, investasi bersama dalam bentuk portofolio efek yang telah terdiversifikasi, dan Manajer Investasi yang dipercaya mengelola dana milik investor.

2.2. Bibit

Bibit adalah aplikasi investasi reksa dana yang membantu investor pemula untuk mulai berinvestasi. Siapa pun bisa langsung investasi dengan optimal sesuai dengan level risiko. Tanpa perlu pengalaman. Tanpa harus bingung. Tinggal terima beres (Bibit, 2020). Berdasarkan kutipan wawancara pada jurnal (Dinandra, 2020) Bibit adalah media untuk jual beli produk reksa dana, dan bukan pemegang uang pengguna. Pada akhir tahun 2019, produk reksa dana yang ada pada aplikasi Bibit sudah mencapai 109 produk reksa dana dari manajer investasi yang berbeda, seperti BNI IDX 30, Schroders, Batavia, Danareksa, dan lain-lain.

2.3. Analisis Sentimen

Analisis sentimen merupakan bagian dari *Natural Language Processing* (NLP) dan *Machine Learning*. Cara kerjanya adalah dengan melakukan klasifikasi terhadap opini positif, negatif, dan netral. Analisis sentimen dalam menganalisis pendapat orang, sentimen, evaluasi, penilaian, sikap, dan emosi terhadap entitas sebagai produk, jasa, organisasi, individu, masalah, peristiwa atau suatu topik (Cahyono, 2017).

Menurut (Mejova, 2009) dikarenakan kompleksitas dari masalah yang diteliti, analisis sentimen dibagi menjadi beberapa *task*.

Task pertama adalah *sentiment or opinion detection*, deteksi sentimen atau pendapat, yang mana mengidentifikasi apakah teks yang diberikan merupakan objektif atau subjektif. Hal ini biasa dilakukan dengan menguji keberadaan kata sifat atau kata keterangan, yang dapat mengidentifikasi apakah teks berisi teks subjektif. Sebagai contoh, kalimat “Ini adalah gambar yang cantik” berisi kata sifat “cantik” yang menunjukkan bahwa teks merupakan teks subjektif.

Task kedua adalah *polarity classification*, klasifikasi polaritas, bertujuan menentukan apakah teks yang diberikan merupakan opini positif, negatif, atau netral. Proses klasifikasi dapat dilakukan dengan cara melakukan *training* pada model *machine learning*. Sebagai contoh, *review* produk yang memberikan label positif atau negatif berdasarkan sentimen yang diungkapkan dalam ulasan.

2.4. Klasifikasi Teks

Klasifikasi merupakan proses penempatan suatu dokumen ke dalam suatu kategori atau kelas sesuai dengan karakteristik dokumen tersebut. Klasifikasi mengacu pada aktivitas menganalisis himpunan dokumen teks *pre-classified* untuk memperoleh suatu model yang dapat digunakan untuk pengelompokan dokumen teks lain yang belum diketahui kelasnya (Nugroho et al., 2003b). Pada saat ini, beberapa jenis metode klasifikasi telah digunakan untuk mengklasifikasikan teks. Ada tiga metode klasifikasi utama yang digunakan; yang pertama adalah klasifikasi teks menggunakan metode statistik seperti *Naive Bayes (NB)*, *K-Nearest Neighbor (KNN)*, *Category Center Vector*, *Support Vector Machine (SVM)*, dan Model Entropi Maksimum. Metode kedua adalah klasifikasi teks berdasarkan koneksi, seperti ketika menggunakan *Artificial Neural Network (ANN)*. Dan metode ketiga adalah klasifikasi berbasis aturan (*rule-based*) seperti *Decision Tree*.

Secara umum langkah-langkah yang biasa dilakukan dalam klasifikasi teks adalah sebagai berikut:

1. Definisikan domain *dataset*

Pengumpulan *dataset* yang mencakup suatu dokumen seperti; *dataset* ulasan aplikasi, *dataset* komentar pada media sosial dan lain sebagainya.

2. *Pre-processing*

Proses awal yang dilakukan untuk membuat data mentah menjadi siap untuk diolah. Proses yang dilakukan seperti membersihkan data dari *noise* dan data-data yang tidak penting.

3. *Transformation*

Merupakan tahapan proses representasi angka yang dihitung dari data teks. Menghitung berapa kali sebuah kata muncul di dalam suatu dokumen digunakan sebagai skema pembobotan dari data teks. Proses yang umum digunakan yaitu *Term-Frequency Inverse Document Frequency* (TF-IDF), *Count Vectorizer* dan lain sebagainya.

4. *Feature Selection*

Penggunaan fitur (*Feature Selection*) dapat membuat klasifikasi menjadi lebih efektif atau efisien dengan mengurangi jumlah data yang dibutuhkan untuk analisis dan dengan mengidentifikasi fitur relevan yang akan digunakan nantinya.

5. *Classification*

Pada tahap ini proses klasifikasi dilakukan dengan menggunakan metodologi klasifikasi tertentu yang sesuai. Beberapa metode yang biasa digunakan untuk mengklasifikasikan data termasuk *Naive Bayes*, *Support Vector Machine*, dan teknik serupa lainnya.

6. *Interpretation/Evaluation*

Tahapan evaluasi yang dilakukan biasanya seperti menghitung akurasi, *recall*, dan *precision* dari metode klasifikasi yang digunakan.

2.5. *Web Scraping*

Web Scraping adalah tindakan mengambil dokumen semi-terstruktur dari halaman *web* dengan tujuan mengekstraksi semua atau sebagian informasi dari halaman internet untuk digunakan untuk kepentingan lain (Prayitno & Rifqi Ma'arif, 2018). Atau berdasarkan pendapat lain dari (Mufid et al., 2020) cara kerja *scraping* sendiri adalah dengan mengakses halaman *web* kemudian menentukan elemen yang nantinya akan dilakukan proses *scraping*. Data yang diperoleh dari proses *scraping* dapat langsung diekstrak atau dapat dimodifikasi sesuai dengan yang dibutuhkan oleh sistem.

Proses *web scraping* pertama kali dilakukan dengan cara disalin secara manual dari halaman *web* ke komputer. Namun, karena butuh banyak waktu dan tenaga untuk menyalin data, cara ini kurang efektif jika ingin mengakses data dalam jumlah besar. Sekarang ini

proses *web scraping* dapat dilakukan secara otomatis dengan berbagai cara seperti dengan membangun perangkat lunak, atau *plugin* browser. Sedangkan pada penelitian ini akan menggunakan metode otomatis yaitu membangun perangkat lunak sederhana menggunakan *library python*.

2.6. Text Pre-processing

Text pre-processing merupakan proses yang dilakukan untuk mempersiapkan data sebelum dilakukannya proses klasifikasi. Tujuan dilakukannya *pre-processing* adalah untuk membersihkan dan penyeragaman data sehingga data tersebut siap untuk di ekstraksi ke tahap selanjutnya. Proses *text pre-processing* secara umum adalah sebagai berikut:

2.6.1. Cleansing

Cleansing merupakan tahapan yang dilakukan untuk menghilangkan *noise* atau elemen-elemen yang tidak dibutuhkan dalam suatu dokumen dengan tujuan untuk memperbaiki dan meningkatkan kualitas data (Himawan et al., 2018). Dalam proses *cleansing* ada beberapa langkah yang dilakukan kepada dokumen seperti menghilangkan semua simbol dan tanda baca, menghapus kesalahan penulisan seperti spasi berlebih. Contoh dari proses *cleansing* dapat dilihat pada tabel 2.1.

Tabel 2.1 Contoh *Cleansing*

sebelum	sesudah
Aplikasi berinvestasi yg bagus bgt buat pemula!. Apalagi buat kita para milenial yg mulai melek finansial.	Aplikasi berinvestasi yg bagus bgt buat pemula Apalagi buat kita para milenial yg mulai melek finansial

2.6.2. Word Normalization

Tahap *word normalization* adalah tahapan mengubah kata-kata yang tidak baku, singkatan, dan kesalahan ketikan yang sering terjadi menjadi bentuk kata baku. Contoh dari proses *word normalization* dapat dilihat pada tabel 2.2.

Tabel 2.2 Contoh *Word Normalization*

sebelum	sesudah
Aplikasi berinvestasi yg bagus bgt buat pemula Apalagi buat kita para milenial yg mulai melek finansial	Aplikasi berinvestasi yang bagus banget buat pemula Apalagi buat kita para milenial yang mulai melek finansial

2.6.3. Tokenizing

Tokenizing merupakan proses yang dilakukan untuk memecah sekumpulan kata dalam sebuah teks menjadi satuan kata, dengan tujuan untuk mendapatkan fitur dari setiap dokumen setelah diberikan bobot nilai. *Tokenizing* berfungsi untuk mengidentifikasi kata

dalam teks menjadi beberapa urutan yang dipotong oleh spasi atau karakter khusus. Kata-kata yang telah dipecah tersebut disebut sebagai token dan digunakan untuk menghitung frekuensi setiap kata dan digunakan dalam proses klasifikasi. Contoh dari proses *tokenizing* dapat dilihat pada tabel 2.3.

Tabel 2.3 Contoh *Tokenizing*

sebelum	sesudah
Aplikasi berinvestasi yang bagus banget buat pemula Apalagi buat kita para milenial yang mulai melek finansial	['Aplikasi', 'berinvestasi', 'yang', 'bagus', 'banget', 'buat', 'pemula', 'Apalagi', 'buat', 'kita', 'para', 'milenial', 'yang', 'mulai', 'melek', 'finansial']

2.6.4. *Casefolding*

Tahap *Casefolding* merupakan proses mengganti semua kata yang telah di *tokenizing* sebelumnya menjadi dalam bentuk huruf kecil. Contoh dari proses *casefolding* dapat dilihat pada tabel 2.4.

Tabel 2.4 Contoh *Casefolding*

sebelum	sesudah
['Aplikasi', 'berinvestasi', 'yang', 'bagus', 'banget', 'buat', 'pemula', 'Apalagi', 'buat', 'kita', 'para', 'milenial', 'yang', 'mulai', 'melek', 'finansial']	['aplikasi', 'berinvestasi', 'yang', 'bagus', 'banget', 'buat', 'pemula', 'apalagi', 'buat', 'kita', 'para', 'milenial', 'yang', 'mulai', 'melek', 'finansial']

2.6.5. *Stopword Removal*

Menurut tahap *stopword removal* adalah tahapan membuang kata-kata yang tidak berpengaruh (Prayitno & Rifqi Ma'arif, 2018) terhadap proses klasifikasi seperti kata depan, kata sambung, dan lain sebagainya. Kata-kata yang termasuk *stopword* tersebut tidak mempunyai pengaruh dan tidak mempunyai nilai pada dokumen teks tersebut. Sehingga kata-kata yang termasuk dalam *stopword* dihilangkan sehingga ukuran pada data juga dapat berkurang. Contoh kata-kata yang termasuk dalam *stopword* dalam Bahasa Indonesia adalah “untuk”, “kapan”, “yang”, “di”, “iya”, “itu”, dan lain sebagainya. Contoh dari *stopword removal* dapat dilihat pada tabel 2.5.

Tabel 2.5 Contoh *Stopword Removal*

sebelum	sesudah
['aplikasi', 'berinvestasi', 'yang', 'bagus', 'banget', 'buat', 'pemula', 'apalagi', 'buat', 'kita', 'para', 'milenial', 'yang', 'mulai', 'melek', 'finansial']	['aplikasi', 'berinvestasi', 'bagus', 'banget', 'pemula', 'apalagi', 'kita', 'para', 'milenial', 'mulai', 'melek', 'finansial']

2.6.6. Stemming

Teknik *stemming* dilakukan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen, dan juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun dalam bentuk yang berbeda karena mendapatkan imbuhan yang berbeda. Contoh *stemming* dapat dilihat pada tabel 2.6.

Tabel 2.6 Contoh *Stemming*

sebelum	sesudah
['aplikasi', ' berinvestasi ', 'bagus', 'banget', 'pemula', 'apalagi', 'kita', 'para', 'milenial', 'mulai', 'melek', 'finansial']	['aplikasi', 'investasi', 'bagus', 'banget', 'pemula', 'apalagi', 'kita', 'para', 'milenial', 'mulai', 'melek', 'finansial']

2.7. Term Frequency-Inverse Document Frequency (TF-IDF)

Term *Frequency-Inverse Document Frequency* atau TF-IDF merupakan suatu metode yang digunakan dalam melakukan pembobotan terhadap kemunculan kata dalam suatu dokumen. *Term Frequency* (TF) menyatakan jumlah kata yang muncul dalam suatu dokumen dan *Inverse Document Frequency* (IDF) menunjukkan tingkat kepentingan suatu kata yang terdapat dalam kumpulan dokumen. Pada TF-IDF terdapat rumus untuk menghitung bobot (W) masing-masing dokumen terhadap kata kunci yang diformulasikan pada persamaan 2.1 dan 2.2.

$$w_{ij} = tf \times idf \dots\dots\dots(2.1)$$

$$w_{ij} = tf \times \log \left(\frac{D}{df} \right) + 1 \dots\dots\dots(2.2)$$

Keterangan :

w_{ij} : bobot kata t_j terhadap dokumen d_i

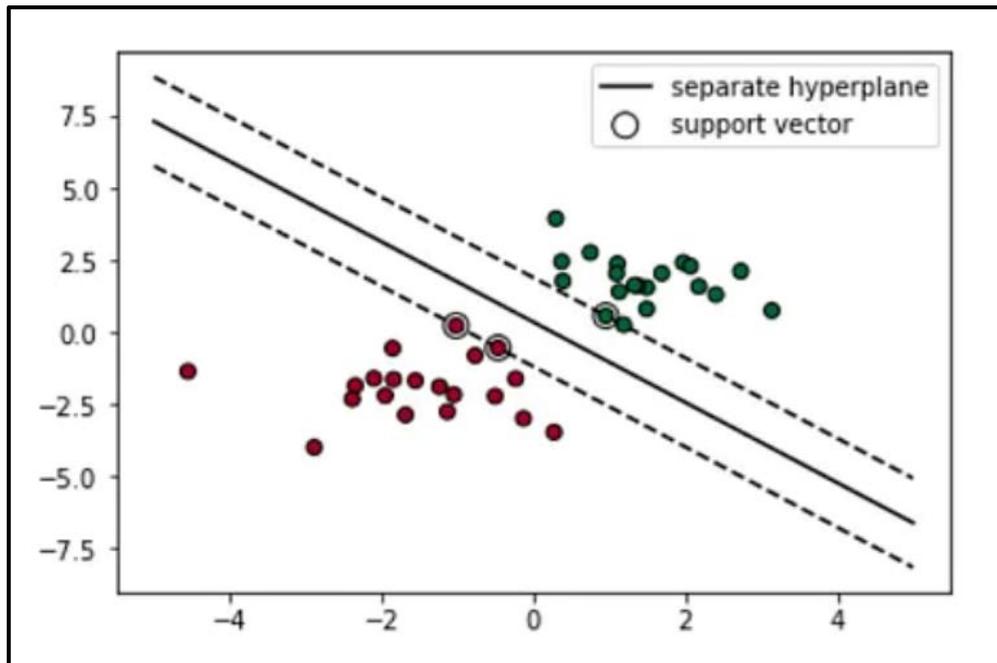
tf_{ij} : bobot kemunculan t_j dalam d_i

D : jumlah semua dokumen yang ada dalam kumpulan dokumen

df : jumlah dokumen yang mengandung kata t_j (minimal terdapat satu kata yaitu term t_j)

2.8. Support Vector Machine

Menurut (Buntoro, 2017) *Support Vector Machine* (SVM) adalah seperangkat metode pembelajaran terbimbing yang menganalisis data dan mengenali pola, digunakan untuk klasifikasi dan analisis regresi. Algoritma SVM dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya (Nugroho et al., 2003a). SVM bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*.



Gambar 2.1 Ilustrasi Metode *Support Vector Machine*

Pada gambar 2.1 memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* -1 untuk merah dan +1 untuk hijau. Yang memperlihatkan bahwa SVM berusaha untuk menemukan *hyperplane* terbaik yang memisahkan kedua *class*. *Hyperplane* pemisah terbaik dapat ditemukan dengan mengukur dan mencari titik maksimal dari margin *hyperplane* tersebut. Margin adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang terdekat ini disebut sebagai *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM (Wibawa et al., 2018).

Dalam kasus data yang bisa dipisahkan secara linear, SVM berusaha mencari *hyperplane* yang memaksimalkan margin, dengan syarat kedua kelas diklasifikasikan dengan benar. Namun pada kenyataannya, kumpulan data mungkin tidak pernah dapat dipisahkan secara linear, sehingga kondisi 100% yang diklasifikasikan dengan benar oleh *hyperplane* tidak akan pernah terpenuhi. Sehingga SVM menangani kasus yang tidak dapat dipisahkan secara linear dengan memperkenalkan dua konsep: *Soft Margin* dan *Kernel Tricks* (Dhiyo et al., 2020).

Dalam penggunaan *soft margin*, SVM menolerir beberapa titik untuk salah klasifikasi dan mencoba untuk menyeimbangkan kompromi antara menemukan garis yang memaksimalkan margin dan meminimalkan kesalahan klasifikasi. Menemukan nilai toleransi yang akan diberikan merupakan parameter yang sangat penting dalam *Support Vector Machine*, baik linear maupun non-linear.

Sedangkan untuk *kernel trick* yang dilakukan adalah memanfaatkan fitur yang ada, menerapkan beberapa transformasi, dan membuat fitur baru. Fitur-fitur baru tersebut adalah kunci bagi SVM untuk menemukan batas keputusan non-linear.

Menurut (Sembiring, 2007) fungsi kernel yang umum digunakan adalah sebagai berikut:

1. Kernel *Linear*

$$K(x_i, x) = x_i^T x \dots \dots \dots (2.3)$$

Keterangan:

x_i = Data latih

x = Data uji

2. Kernel *Polynomial*

$$K(x_i, x) = (\gamma \cdot x_i^T x + r)^p, \gamma > 0 \dots \dots \dots (2.4)$$

Keterangan:

x_i = Data latih

x = Data uji

p = Derajat Polynomial

r = Nilai Konstanta

γ = *Scaling parameter* dari jarak *Euclidan*

3. Kernel *Radial Basis Function* (RBF)

Radial Basis Function (RBF) merupakan salah satu kernel yang digunakan dalam pengklasifikasian data non-linear. Menurut (Mase et al., 2018) kernel RBF bisa dianggap sebagai transformator atau prosesor untuk menghasilkan fitur baru dengan mengukur jarak antara semua titik lainnya ke pusat titik/titik tertentu.

$$K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2), \gamma > 0 \dots \dots \dots (2.5)$$

Keterangan:

x_i = Data latih

x_j = Data uji

γ = *Scaling parameter* dari jarak *Euclidan*

4. Kernel *Sigmoid*

$$K(x_i, x) = \tan(\gamma \cdot x_i^T x + r) \dots \dots \dots (2.6)$$

Keterangan:

x_i = Data latih

x = Data uji

r = Nilai Konstanta

γ = *Scaling parameter* dari jarak *Euclidan*

2.9. Confusion Matrix

Confusion matrix adalah sebuah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan prediksi yang dilakukan oleh model dengan nilai sebenarnya dari data yang diamati. Performa suatu model klasifikasi dapat dilihat dari parameter pengukuran performa yaitu tingkat akurasi, *recall*, dan presisi. Struktur persegi dari *confusion matrix* direpresentasikan melalui baris dan kolom, di mana baris merupakan kelas sebenarnya, dan kolom adalah kelas yang diprediksi. Untuk klasifikasi biner, *confusion matrix* direpresentasikan dalam matriks 2*2 sedangkan untuk klasifikasi *multiclass* dengan kelas k, akan menggunakan matriks k*k (Hasnain et al., 2020). Terdapat empat istilah yang digunakan dalam merepresentasikan hasil dari proses klasifikasi pada *confusion matrix*, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Tabel *confusion matrix* dapat dilihat pada tabel 2.6.

Tabel 2.7 *Confusion Matrix*

		Prediksi	
		Negatif	Positif
Aktual	Negatif	<i>True negative</i> (TN)	<i>False positive</i> (FP)
	Positif	<i>False negative</i> (FN)	<i>True positive</i> (TP)

Keterangan:

TP : jumlah prediksi benar bahwa yang diprediksi hasilnya positif

FP : jumlah prediksi salah yang seharusnya bernilai negatif diprediksi positif

TN : jumlah prediksi benar bahwa hasil prediksi bernilai negatif

FN : jumlah prediksi salah, seharusnya bernilai positif diprediksi negatif

Berdasarkan tabel *confusion matrix* tersebut dapat menghitung *performance matrix* untuk mengukur model yang telah dibuat. *Performance matrix* yang sering digunakan adalah akurasi, *recall*, dan presisi.

1. Akurasi

Akurasi merupakan nilai yang merepresentasikan tingkat kedekatan antara nilai prediksi dengan nilai aktual. Perhitungannya dilakukan dengan cara membagi jumlah data yang diklasifikasikan secara benar dengan total data pelatihan (*tested data*). Persamaan untuk menghasilkan nilai akurasi yaitu dengan membandingkan nilai prediksi dengan keseluruhan data, seperti pada persamaan berikut.

$$accuracy = \frac{TP}{Total\ Data} \times 100\% \dots \dots \dots (2.4)$$

2. Recall

Recall merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Persamaan recall ditunjukkan pada persamaan berikut.

$$recall = \frac{TP}{TP+FN} \times 100\% \dots \dots \dots (2.5)$$

3. Presisi

Presisi merupakan metode pengujian dengan melakukan perbandingan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi yang telah terambil oleh sistem baik relevan maupun tidak. persamaan presisi seperti berikut.

$$precision = \frac{TP}{TP+FP} \times 100\% \dots \dots \dots (2.6)$$

2.10. Studi Pustaka

Tabel 2.8 Tabel Studi Pustaka

No	Penulis	Judul	Inti sari
1	(Muis & Affandes, 2015)	Penerapan Metode <i>Support Vector Machine</i> (SVM) Menggunakan Kernel <i>Radial Basis Function</i> (RBF) Pada Klasifikasi <i>Tweet</i>	Penelitian ini menggunakan metode <i>Support Vector Machine</i> (SVM) dengan kernel RBF untuk mengklasifikasikan data <i>tweet</i> menjadi iklan atau bukan iklan. Pengujian menunjukkan akurasi stabil dengan rentang nilai parameter C ($0 \leq C \leq 3$ dan $0.01 \leq \gamma \leq 10$) pada data tanpa pemilihan fitur, dan rentang nilai C ($0 \leq C \leq 300$ dan $0.01 \leq \gamma \leq 10$) setelah pemilihan fitur. Hasil ini dapat digunakan untuk membantu pengguna Twitter memfilter <i>tweet</i> iklan pada akun mereka.
2	(Himawan et al., 2018)	Metode <i>Lexicon Based</i> dan <i>Support Vector Machine</i> Untuk Menganalisis Sentimen Pada Media Sosial Sebagai Rekomendasi Oleh - Oleh Favorit.	penelitian ini fokus pada analisis sentimen pengguna media sosial terhadap oleh-oleh favorit di daerah Yogyakarta. Analisis sentimen dilakukan menggunakan algoritma <i>Lexicon Based</i> dan <i>Support Vector Machine</i> (SVM) dengan data dari Twitter dan Instagram. Data latih terdiri dari 1000 sentimen, sementara data uji terdiri dari 50 sentimen. Hasil pengujian menunjukkan akurasi tertinggi menggunakan metode <i>Lexicon Based</i> sebesar 87.78%, presisi tertinggi sebesar 94.23% menggunakan <i>Lexicon Based</i> , dan <i>recall</i> tertinggi sebesar 100% menggunakan SVM.
3	(Ardhani et al., 2021)	<i>Sentiment Analysis Towards Kartu Prakerja Using Text Mining with Support Vector Machine and</i>	Penelitian ini berfokus terhadap penggunaan kernel <i>Radial Basis Function</i> (RBF) pada metode SVM pada penelitian terhadap analisis sentimen kartu prakerja. Dengan data sebanyak 500 <i>tweet</i> , 60% bersifat sentimen negatif dan 40% bersifat sentimen positif. Pengklasifikasian pada data tes menampilkan akurasi rata-rata sebesar 85.20%,

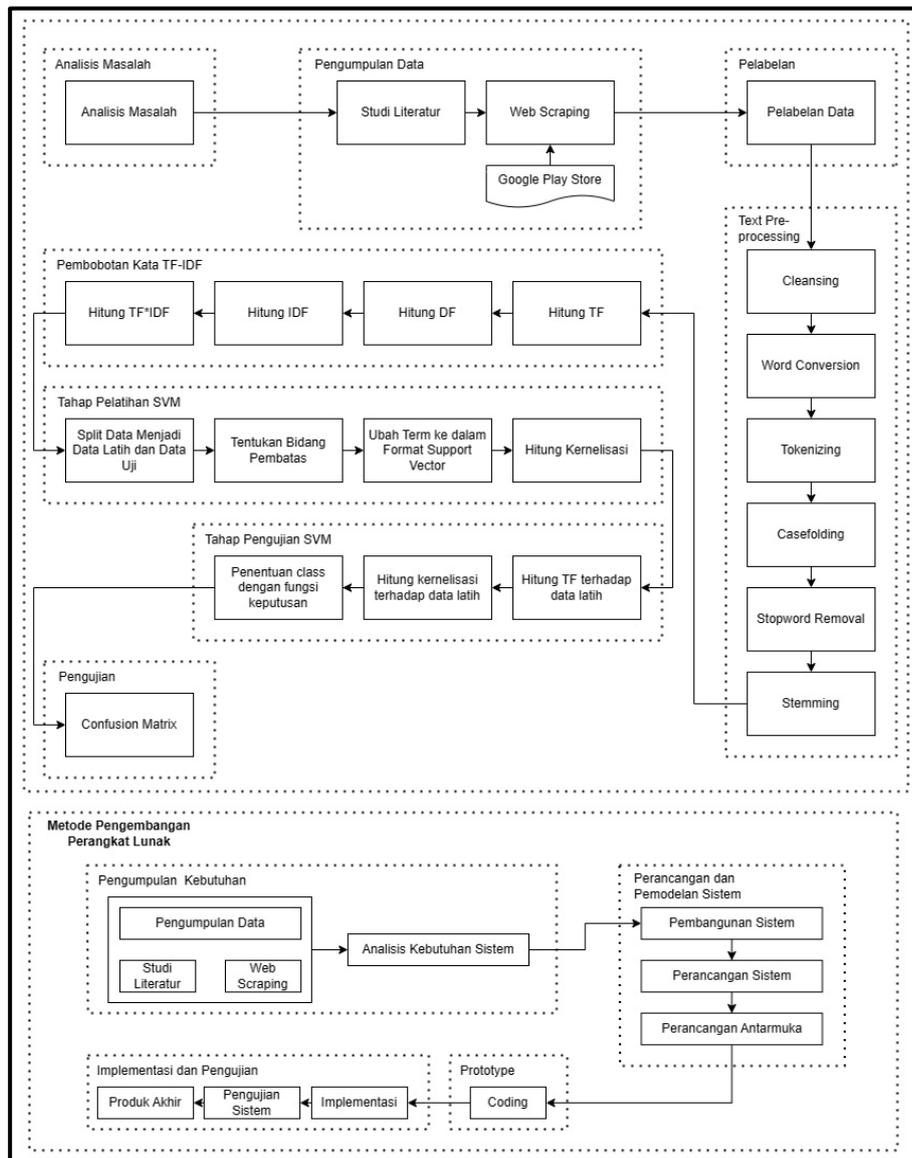
		<i>Radial Basis Function Kernel</i>	sensitivitas sebesar 91.68%, kespesifikan sebesar 75.75%, prediksi sentimen positif sebesar 85.03% dan sentimen negatif sebesar 86.04%.
4	(Rachman & Purnami, 2012)	Perbandingan Klasifikasi Tingkat Keganasan <i>Breast Cancer</i> Dengan Menggunakan Regresi Logistik Ordinal Dan <i>Support Vector Machine</i> (SVM)	Penelitian ini menggunakan analisis regresi logistik ordinal dan <i>Support Vector Machine</i> (SVM) untuk menganalisis tingkat keganasan <i>breast cancer</i> . Hasil penelitian menggunakan regresi logistik ordinal menunjukkan bahwa variabel yang berpengaruh terhadap tingkat keganasan <i>breast cancer</i> adalah ukuran tumor dan nodus, dengan tingkat ketepatan klasifikasi tertinggi sebesar 56,60%. Sementara itu, ketepatan klasifikasi tertinggi menggunakan SVM dengan kernel RBF dan <i>polynomial</i> mencapai 98,11%.
5	(Indraswari & Arifin, 2017)	<i>RBF Kernel Optimization Method with Particle Swarm Optimization on SVM Using the Analysis of Input Data's Movement</i>	Penelitian ini menggunakan metode optimasi <i>Particle Swarm Optimization</i> pada kernel RBF untuk metode SVM. Hasil eksperimen pada <i>dataset Breast Cancer Wisconsin</i> memperlihatkan bahwa metode optimasi kernel RBF dapat meningkatkan akurasi pada SVM secara signifikan. Metode ini memiliki tingkat kompleksitas yang lebih rendah dibandingkan dengan metode optimasi SVM lainnya yang menghasilkan <i>running time</i> yang lebih cepat.

BAB III

METODE PENELITIAN DAN PENGEMBANGAN SISTEM

3.1. Metode Penelitian

Metode penelitian yang digunakan adalah metode penelitian kuantitatif. Menurut buku yang ditulis oleh (Ahyar & Sukmana, 2020) penelitian kuantitatif memiliki arti penelitian yang banyak menggunakan angka, dimulai dari proses pengumpulan data, analisis data, dan penampilan data. Metode penelitian kuantitatif menekankan analisis pada numerik (angka) yang kemudian dianalisis dengan metode statistik yang sesuai. Data yang digunakan pada penelitian kuantitatif tidak harus berupa data kuantitatif, tetapi juga dapat menggunakan data kualitatif maupun gabungan dari keduanya. Adapun tahapan dari penelitian yang dilakukan dapat dilihat pada gambar 3.1.



Gambar 3.1 Metode Penelitian Kuantitatif

Secara garis besar penelitian ini dilakukan dalam beberapa tahapan yaitu, pengumpulan data, pelabelan, *text pre-processing*, pembobotan kata TF-IDF, tahap pelatihan SVM, tahap Pengujian SVM, dan pengujian.

3.1.1. Analisis Masalah

Analisis masalah merupakan suatu gambaran masalah yang diangkat dalam penulisan tugas akhir Analisis Sentimen Aplikasi Investasi Reksa Dana Bibit Menggunakan Metode *Support Vector Machine*. Penelitian ini dilakukan untuk merancang sebuah sistem yang dapat memperlihatkan tingkat akurasi dan presisi dari metode *Support Vector Machine* yang digunakan dalam penelitian ini.

Salah satu cara yang dapat digunakan adalah dengan melakukan klasifikasi teks. Dengan menggunakan data berupa ulasan yang diberikan oleh pengguna aplikasi Bibit melalui halaman *review* di *google play store*.

3.1.2. Pengumpulan Data

Data merupakan salah satu komponen paling penting dalam suatu penelitian. Penelitian memerlukan data untuk diolah sehingga menjadi sebuah informasi yang dapat mendukung penelitian yang akan dilakukan. Pengumpulan data yang dilakukan dalam penelitian ini adalah studi literatur, dan *web scraping*.

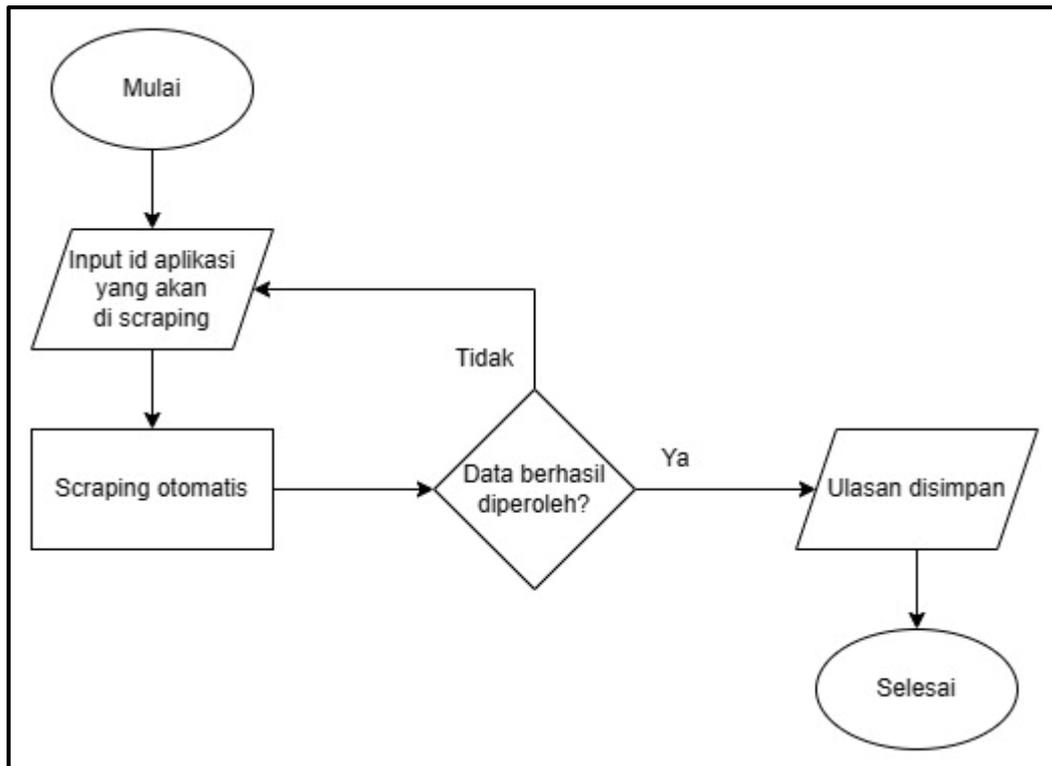
3.1.2.1. Studi literatur

Studi literatur merupakan tahapan yang dilakukan untuk memperoleh informasi yang berkaitan dengan masalah penelitian, metode serta kelebihan dan kekurangan pada penelitian-penelitian sebelumnya. Studi literatur dilakukan sebagai referensi untuk mendukung penelitian ini. Studi literatur bersumber dari buku teks, jurnal ilmiah dan sumber yang dapat dipertanggungjawabkan.

3.1.2.2. Web scraping

Pada bagian ini dilakukan pengumpulan data teks ulasan aplikasi Bibit yang dilakukan dengan proses *web scraping*. *Web scraping* dilakukan untuk mengumpulkan data ulasan aplikasi Bibit dari *google play store*. Proses *web scraping* dilakukan dengan pemrograman *software* menggunakan *python* dengan bantuan *library google_play_scraper* untuk mengumpulkan data ulasan dari *google play store*.

Berikut *flowchart web scraping* yang merupakan gambaran alur dari mengambil data ulasan, seperti yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram *flowchart web scraping*

Seperti yang telah dijelaskan di atas, proses *web scraping* dilakukan dengan cara pemrograman *software* menggunakan *python* dengan bantuan *library google_play_scraper* untuk mengumpulkan data ulasan dari *google play store*. Proses ini dimulai dengan melakukan input id dari aplikasi Bibit yang menjadi sumber ulasan pada penelitian ini. Selanjutnya akan dilakukan proses *scraping* otomatis. Jika dalam proses *scraping* data tidak diperoleh, maka akan dilakukan input ulang pada id aplikasi yang dituju. Data yang didapatkan dari proses *web scraping* akan disimpan dalam *file* dengan ekstensi *.csv* seperti gambar berikut.

	reviewId	userName	userImage	content	score
1	d76fd51d-6ba6-462d-...	Syahrul kaka	https://play-lh.google...	Aplikasinya bagus unt...	4
2	d263f403-e8e8-42f1-...	Dana Dyaksa	https://play-lh.google...	terimakasih bibit, apli...	5
3	2d9c25d0-c510-44c2-...	Elga Rian Ardiyansa	https://play-lh.google...	Sudah saatnya bibit ...	4
4	a053b59c-31f4-4d10-...	Zni20	https://play-lh.google...	kecewa sama bibit,, k...	2
5	dbf3602f-368d-466e-...	Rio Budi harto	https://play-lh.google...	untuk aplikasi sangat ...	5
6	aa764f2a-1e35-44ad-...	Khadijah Yusuf	https://play-lh.google...	Mau tambah rekening...	3
7	2c2e0b2f-201f-401d-...	tri astuti	https://play-lh.google...	untuk melakukan penj...	3
8	2ae81bee-a104-4546-...	Devi septiyani	https://play-lh.google...	Cocok buat anak mud...	5
9	de18fde5-afaf-46d5-9...	Luqman Zuhdi	https://play-lh.google...	Selama saya pakai b...	5
10	607fa46d-a126-4a32-...	Hilal Fj	https://play-lh.google...	Bagus buat investasi j...	4

Gambar 3.3 Hasil *web scraping*

3.1.3. Pelabelan Data

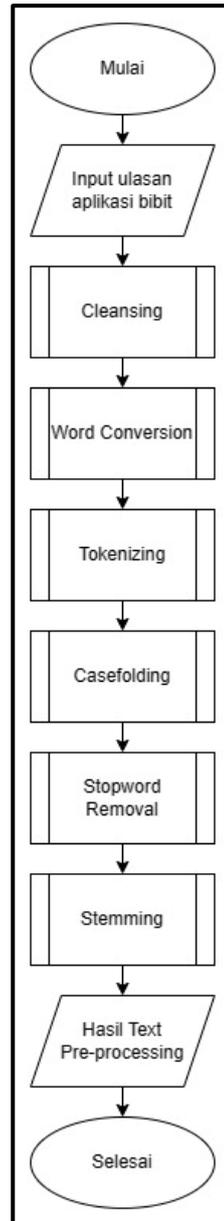
Data yang telah terkumpul selanjutnya akan dilakukan pelabelan secara manual dengan label Komentar untuk ulasan pengguna Bibit dan Sentimen untuk label positif dan negatif. Proses pelabelan data dilakukan secara manual dengan cara melakukan pengecekan terhadap komentar yang diberikan pengguna aplikasi Bibit. *Score* yang diberikan oleh pengguna pada halaman *web google play store* dijadikan pedoman tambahan dalam melakukan pelabelan data. Hasil dari tahap pelabelan disajikan dalam tabel 3.1.

Tabel 3.1 Contoh hasil pelabelan data

Komentar	Sentimen
Aplikasi bibit ini ccok banget buat untuk pemula yang mau investasi tapi masih belajar simpel dan mudah gunakannya dan pendaftaran cukup menggunakan ktp dan juga untuk keamanan jangan di ragukan lagi. Untuk pembelian saham dan bitcoin juga harganua murah dan ccok untuk yang mau mulai nyoba	Positif
untuk aplikasi sangat mudah dj gunakan, tetapi kalau bisa proses withdraw di percepat jangan menunggu 3 hari kerja. overall keren si bibit	Positif
Mau tambah rekening bank susah banget proses perekaman wajah nya itu susah dan terlalu capet jadi waktu mau ikutin arahan di aplikasi nya itu susaah baanget tolong dong lebih di permudah lagi.	Negatif
Selama saya pakai bagus aplikasinya, tapi kadang berat waktu awal masuk (stuck agak lama di logo bibit). Sama saya mau ngasih saran barangkali bisa dipertimbangkan : karena kemarin bibit udah tambah fitur beli saham dan fitur beli obligasi fr, mungkin kedepannya bisa ditambah fitur jual beli mata uang asing, emas atau crypto sekalian? Atau bisa jadi Stockbit disatukan ke bibit sekalian biar bibit jadi super app investasi?	Positif
Bagus buat investasi jangka panjang, cuma untuk pembelian dan penjualan prosesnya di cepatkan jangan sampai berhari-hari prosesnya. Kasian jika orang yang urgent tiba-tiba butuh uang dan melakukan penjualan dari sini, tapi proses pengembaliannya berhari-hari.. Overall is good	Positif
Loadingnya lama, mana sering keluar mulu. Kenapa gini? Saya baru mau mulai belajar investasi. Tolong pihak yang bersangkutan, dibenerin lagi apknya. Masa waktu dibuka aplikasinya, sedikit-sedikit keluar	Negatif
Menu data pribadi saya berwarna abu ² dan tidak bisa di klik, sudah saya lakukan reinstal tetap saja. Edit : Sudah bisa menu data pribadinya, thx u min penanganan kurang dari 1 hari kerja	Positif
Yolong di perbaiki. Proses pembelian sangat lama. Jd ga dapet harga yg di inginkan di hari yg sama. Update perubahan harga nya juga lambat. Opsi top up ga bisa lewat minimarket.	Negatif
Aplikasi cukup mudah dimengerti untuk pemula. Tapi saran saja kali ya. Ketika mau buka aplikasinya jangan langsung ke beranda, misal masuk kan pin atau password dulu gitu.	Positif

3.1.4. Text Pre-processing

Text pre-processing dilakukan untuk membersihkan data konten yang menjadi masukan dari sistem. Proses ini dilakukan untuk membuat data sama rata dan siap digunakan. *Text pre-processing* memiliki beberapa tahapan yang menjadi sub-proses di antaranya adalah *cleansing*, *word conversion*, *tokenizing*, *casefolding*, *stopword removal* dan *stemming*. *Flowchart* dari *text pre-processing* dapat dilihat pada gambar 3.4.

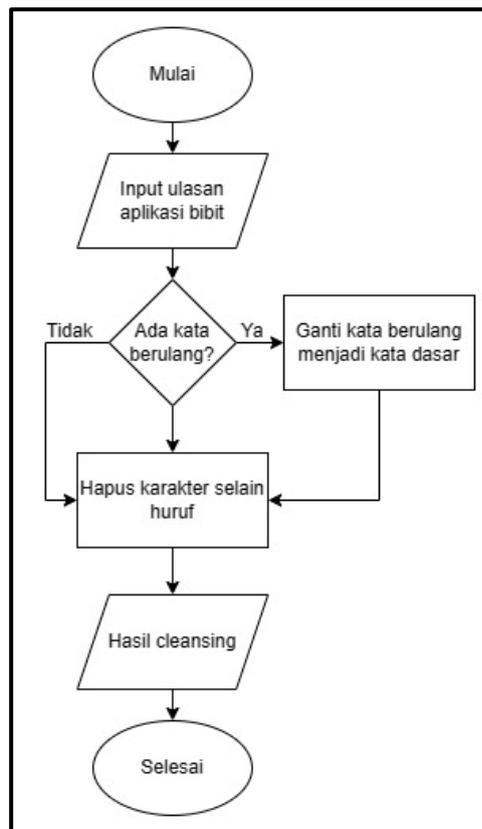


Gambar 3.4 Diagram *flowchart text pre-processing*

3.1.4.1. Flowchart tahap *cleansing*

Tahap *cleansing* merupakan proses yang dilakukan untuk menghilangkan elemen-elemen yang dinilai kurang penting seperti menghilangkan angka dan karakter selain huruf. Proses menghilangkan angka dan karakter selain huruf dilakukan dengan menggunakan

bantuan *library regular expression*. Alur proses dari tahapan *cleansing* dapat dilihat dari *flowchart* tahap *cleansing* seperti gambar 3.5.



Gambar 3.5 Diagram *flowchart cleansing*

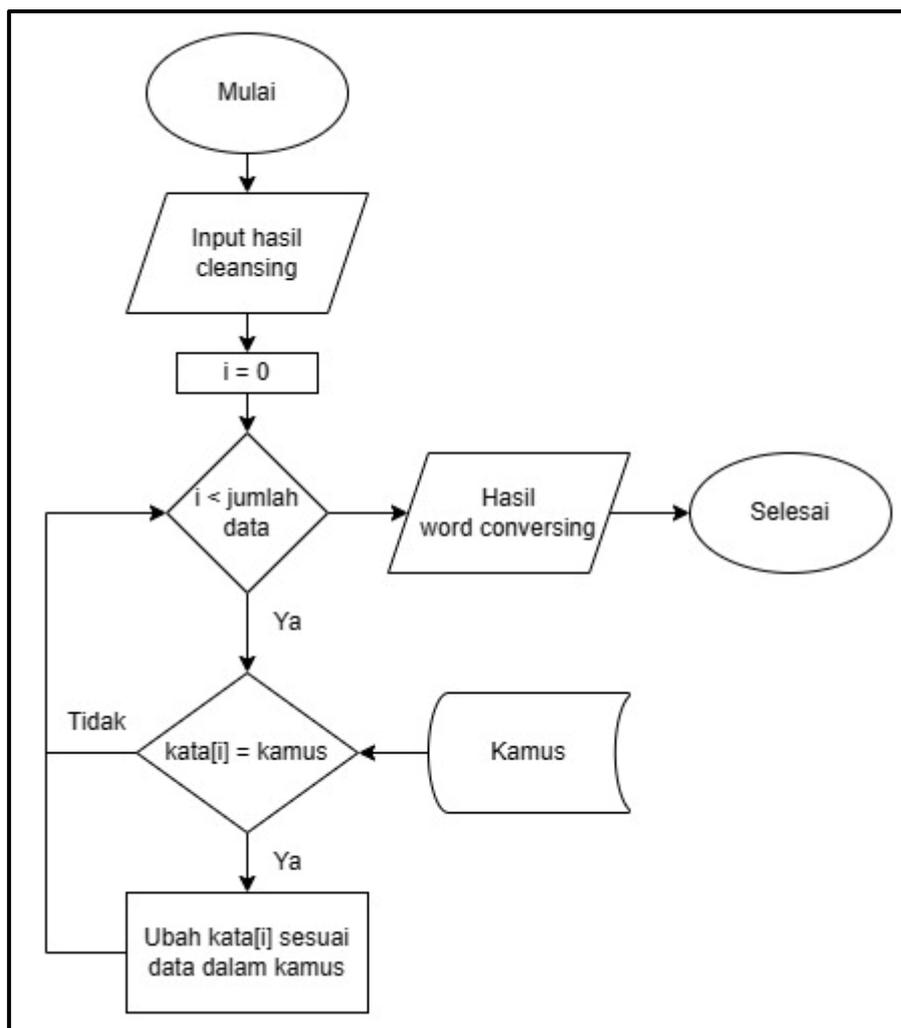
Proses *cleansing* dimulai dengan melakukan input data ulasan dari aplikasi Bibit, jika dari data tersebut ditemukan kata yang berulang maka kata berulang tersebut akan diubah menjadi kata dasar, selanjutnya akan dilakukan penghapusan karakter lain yang bukan merupakan huruf. Setelah proses *cleansing* berhasil dilakukan maka akan didapatkan hasil dari proses *cleansing* tersebut. Contoh hasil *cleansing* dapat dilihat pada tabel 3.2.

Tabel 3.2 Contoh penerapan *cleansing*

Sebelum	Sesudah
Aplikasi berinvestasi yg bagus bgt buat pemula!. Apalagi buat kita para milenial yg mulai melek finansial.	Aplikasi berinvestasi yg bagus bgt buat pemula Apalagi buat kita para milenial yg mulai melek finansial

3.1.4.2. *Flowchart* tahap *word conversion*

Tahap *word conversion* merupakan tahapan mengganti kata-kata tidak baku atau singkatan dari hasil tahapan *cleansing* sebelumnya menjadi kata baku. *Flowchart* dari proses *word conversion* dapat dilihat pada gambar 3.6.



Gambar 3.6 Diagram *flowchart* tahap *word conversion*

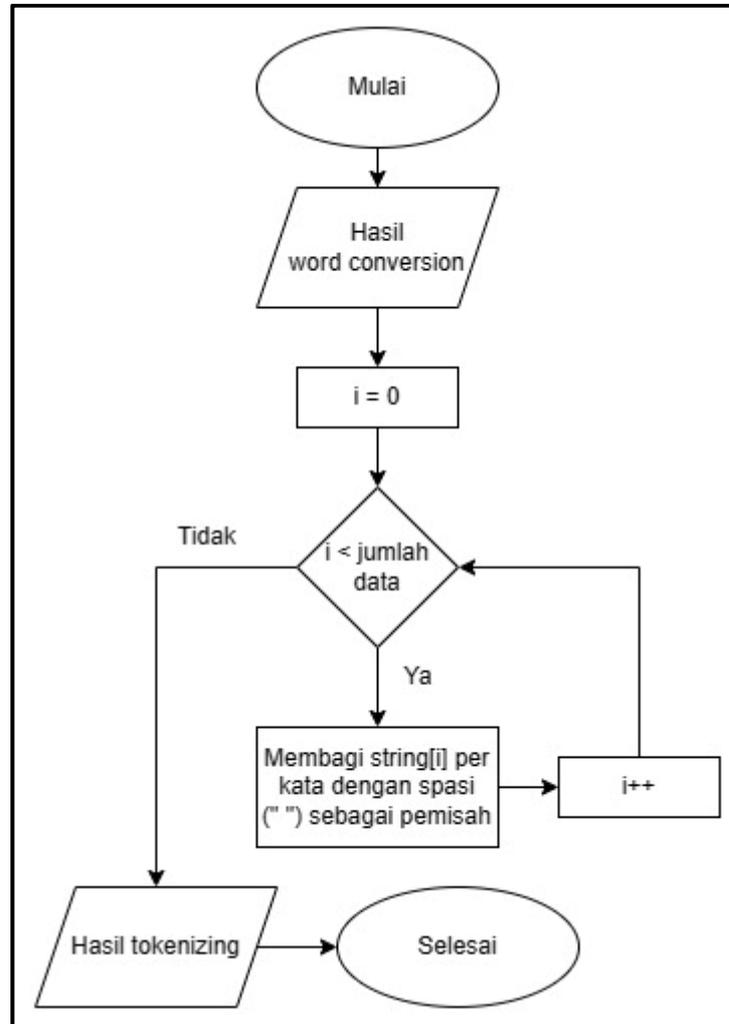
Proses *word conversion* akan menerima input dari hasil *cleansing* sebelumnya. Dalam proses ini akan dilakukan pengecekan per kata dan menggunakan fungsi sinonim dengan mengganti kata yang sebelumnya sudah dibuat daftar kata sinonim dari kata yang tidak baku dan kata singkatan dalam *file* kamus bernama “*kamusalay.csv*”. jika kata tersebut merupakan kata tidak baku atau singkatan seperti yang terdapat dalam kamus, maka akan diganti dengan kata baku. Apabila proses *word conversion* sudah dilakukan terhadap semua kata maka hasil *word conversion* akan disimpan dan siap untuk melakukan tahapan selanjutnya. Berikut contoh penerapan *word conversion* yang dapat dilihat pada tabel 3.3.

Tabel 3.3 Contoh penerapan *word conversion*

Sebelum	Sesudah
Aplikasi berinvestasi yg bagus bgt buat pemula Apalagi buat kita para milenial yg mulai melek finansial	Aplikasi berinvestasi yang bagus sekali buat pemula Apalagi buat kita para milenial yang mulai melek finansial

3.1.4.3. Flowchart tahap tokenizing

Tokenizing adalah proses pemisahan kata dari bentuk *string* pada kalimat. Hasil dari tahap *word conversion* sebelumnya akan dilakukan proses *tokenizing* seperti pada *flowchart* yang dapat dilihat pada gambar 3.7.



Gambar 3.7 Diagram *flowchart* tahap *tokenizing*

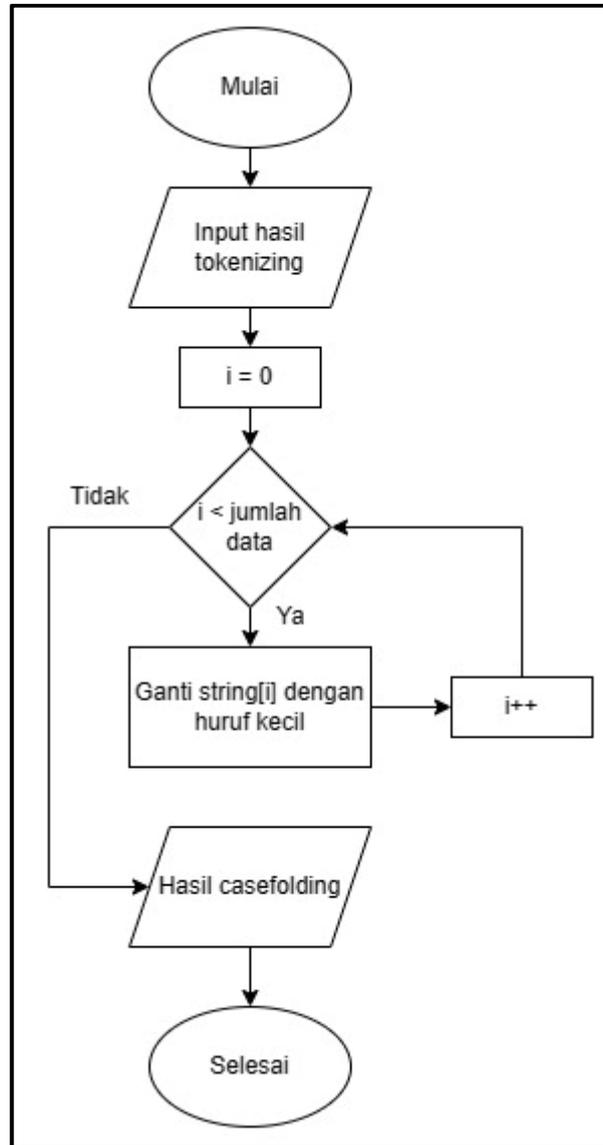
Berikut merupakan contoh penerapan proses *tokenizing* terhadap data hasil *word conversion* sebelumnya yang dapat dilihat pada tabel 3.4.

Tabel 3.4 Contoh penerapan *tokenizing*

Sebelum	Sesudah
Aplikasi berinvestasi yang bagus sekali buat pemula Apalagi buat kita para milenial yang mulai melek finansial	['Aplikasi', 'berinvestasi', 'yang', 'bagus', 'sekali', 'buat', 'pemula', 'Apalagi', 'buat', 'kita', 'para', 'milenial', 'yang', 'mulai', 'melek', 'finansial']

3.1.4.4. Flowchart tahap casefolding

Casefolding merupakan proses mengganti semua kata yang telah di *tokenizing* sebelumnya menjadi dalam bentuk huruf kecil. *Flowchart* tahap *casefolding* dapat dilihat pada gambar 3.8.



Gambar 3.8 Diagram *flowchart* tahap *casefolding*

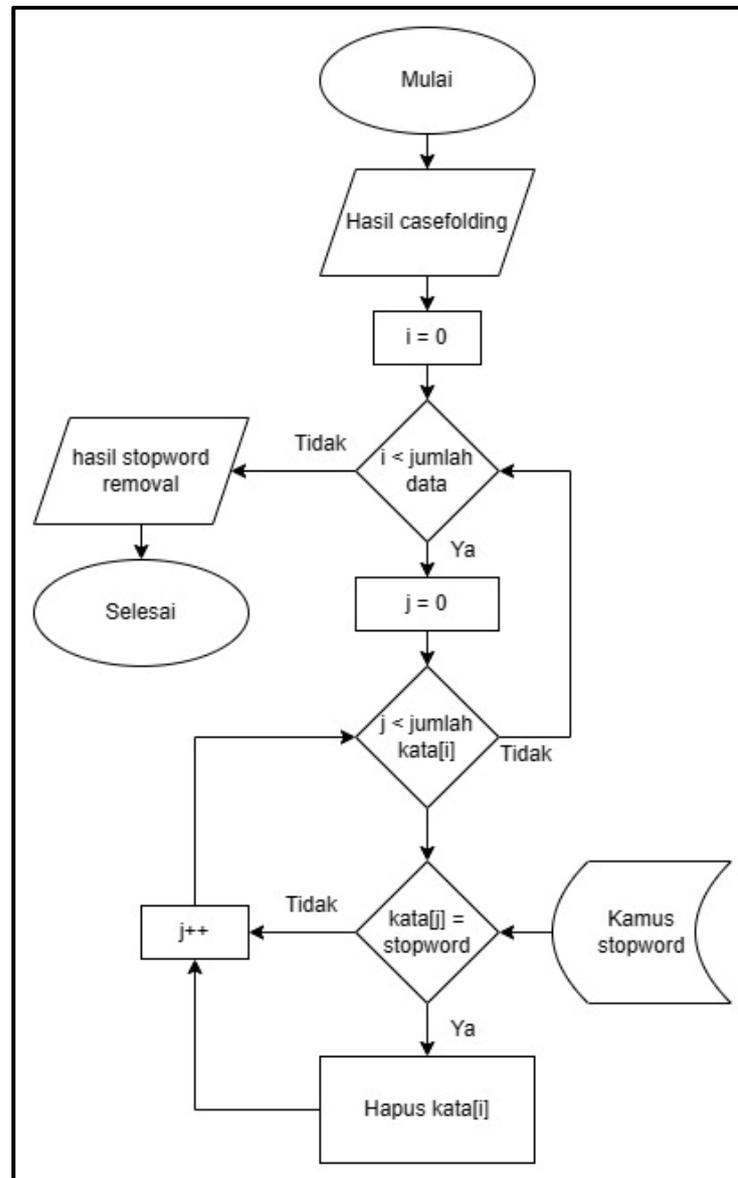
Berikut contoh dari tahap *casefolding* yang dapat dilihat pada tabel 3.5.

Tabel 3.5 Contoh penerapan *casefolding*

Sebelum	Sesudah
['Aplikasi', 'berinvestasi', 'yang', 'bagus', 'sekali', 'buat', 'pemula', 'Apalagi', 'buat', 'kita', 'para', 'milenial', 'yang', 'mulai', 'melek', 'finansial']	['aplikasi', 'berinvestasi', 'yang', 'bagus', 'sekali', 'buat', 'pemula', 'apalagi', 'buat', 'kita', 'para', 'milenial', 'yang', 'mulai', 'melek', 'finansial']

3.1.4.5. Flowchart tahap *stopword removal*

Stopword removal merupakan proses penghapusan kata-kata yang sering muncul dan tidak memiliki makna. *Flowchart* proses *stopword removal* dapat dilihat pada gambar 3.9.



Gambar 3.9 Diagram *flowchart* tahap *stopword removal*

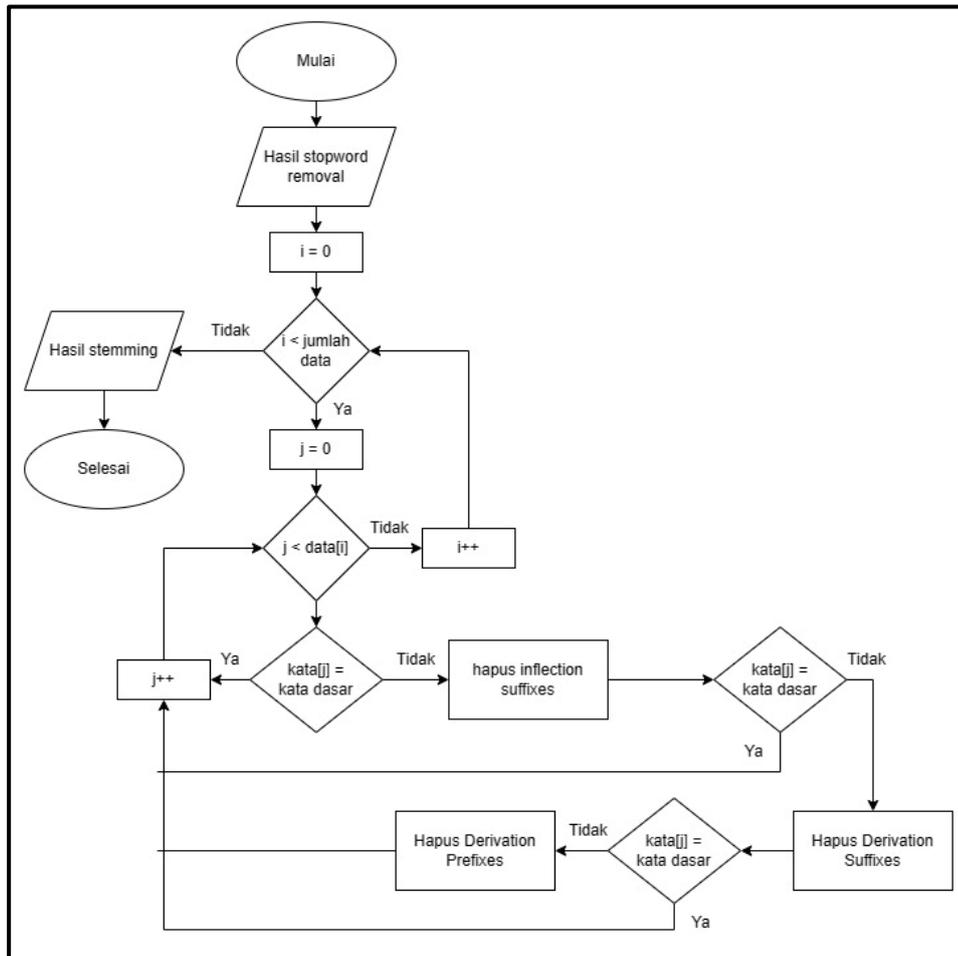
Proses *stopword removal* memproses input berupa hasil proses sebelumnya. Dalam proses ini akan dilakukan pengecekan per kata dan menghapus kata yang tidak memiliki makna. *Stopword removal* dilakukan dengan mengecek kamus *stopword* bahasa Indonesia yang disediakan oleh *library (Natural Language Toolkit) NLTK corpus*. Jika kata yang dicek terdapat dalam kamus, maka kata tersebut akan dihapus. Dan kata yang tidak termasuk dalam kamus akan disimpan. Setelah setiap kata diperiksa, maka akan disimpan. Berikut contoh dari penerapan *stopword removal* seperti pada tabel 3.6.

Tabel 3.6 Contoh penerapan *stopword removal*

Sebelum	Sesudah
['aplikasi', 'berinvestasi', 'yang', 'bagus', 'sekali', 'buat', 'pemula', 'apalagi', 'buat', 'kita', 'para', 'milenial', 'yang', 'mulai', 'melek', 'finansial']	['aplikasi', 'berinvestasi', 'bagus', 'pemula', 'milenial', 'melek', 'finansial']

3.1.4.6. Flowchart tahap stemming

Stemming adalah proses menghilangkan imbuhan dari sebuah kata atau pembentukan kata dasar. *Flowchart* dari proses *stemming* dapat dilihat pada Gambar 3.10.



Gambar 3.10 Diagram *flowchart* tahap *stemming*

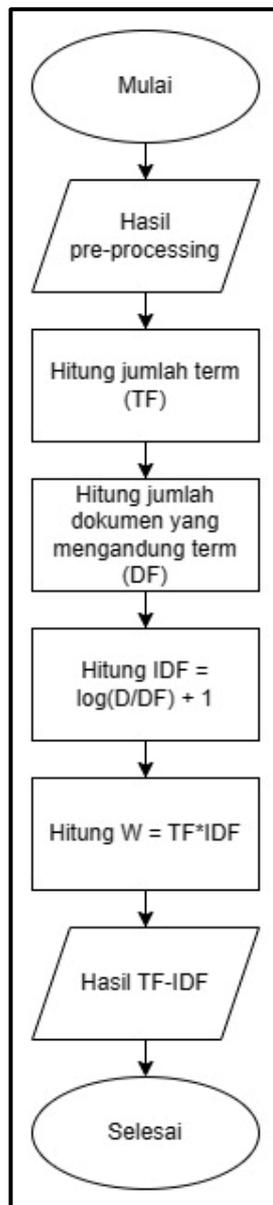
Proses *stemming* akan menerima input berupa hasil proses sebelumnya yaitu *stopword removal*. Dalam proses *stemming* akan dilakukan pengecekan per kata dan mencari kata dasar. Jika kata yang dicek bukan merupakan kata dasar, maka akan dihapus *inflection suffixes*, *derivational suffixes*, dan *derivation prefixes*. Apabila semua kata telah diperiksa, maka hasil *stemming* akan disimpan. Berikut contoh penerapan proses *stemming* seperti pada tabel 3.7.

Tabel 3.7 Contoh penerapan *stemming*

Sebelum	Sesudah
['aplikasi', 'berinvestasi', 'bagus', 'pemula', 'milenial', 'melek', 'finansial']	['aplikasi', 'investasi', 'bagus', 'pemula', 'milenial', 'melek', 'finansial']

3.1.5. Pembobotan Kata TF-IDF

Pembobotan kata merupakan proses yang dilakukan untuk memberikan nilai atau bobot terhadap setiap kata. Pembobotan kata pada penelitian ini menggunakan fitur ekstraksi TF-IDF yang membobotkan kata dengan cara mengalihkan hasil *term frequency* (TF) dengan hasil *inverse document frequency* (IDF). Berikut merupakan diagram *flowchart* proses pembobotan kata TF-IDF seperti pada gambar 3.11.



Gambar 3.11 Diagram *flowchart* TF-IDF

Berikut merupakan contoh analisis pada setiap tahapan pembobotan kata TF-IDF terhadap kata hasil proses *text pre-processing* sebagai berikut. Terdapat lima data contoh yang akan digunakan pada proses pembobotan kata TF-IDF yang sebelumnya telah dilakukan proses *text pre-processing*.

Data 1 : “Cocok untuk pemula dan rekomendasi sekali, terus berikan yg terbaik untuk milenial untuk berinvestasi”

Hasil *Pre-processing* : [‘cocok’, ‘mula’, ‘rekomendasi’, ‘beri’, ‘baik’, ‘milenial’, ‘investasi’] (Positif)

Data 2 : “Aplikasi yang rapi dan mudah digunakan.”

Hasil *Pre-processing* : [‘aplikasi’, ‘rapi’, ‘mudah’, ‘guna’] (Positif)

Data 3 : “Sungguh kecewa dengan pengembalian dananya lama”

Hasil *Pre-processing* : [‘sungguh’, ‘kecewa’, ‘kembali’, ‘dana’, ‘lama’] (Negatif)

Data 4 : “*Friendly user* dan tampilan yang enak banget. Mudah di gunakan oleh pemula.”

Hasil *Pre-processing* : [‘friendly’, ‘user’, ‘tampil’, ‘enak’, ‘mudah’, ‘guna’, ‘pemula’] (Positif)

Data 5 : “Mau *ngecek* tapi kenapa *nggak* bisa dibuka??”

Hasil *Pre-processing* : [‘cek’, ‘tidak’, ‘bisa’, ‘buka’] (Negatif)

3.1.5.1. Perhitungan TF

Perhitungan *term frequency* (TF) dilakukan untuk mendapatkan jumlah setiap kata atau fitur yang ada di masing-masing dokumen. Perhitungan *term frequency* dilakukan dengan menghitung kemunculan setiap kata dalam dokumen. Hasil dari perhitungan TF dapat dilihat pada tabel 3.8 tabel 3.9.

Tabel 3.8 Perhitungan TF

Indeks	Term	TF				
		D1	D2	D3	D4	D5
0	cocok	1	0	0	0	0
1	mula	1	0	0	1	0
2	rekomendasi	1	0	0	0	0
3	beri	1	0	0	0	0
4	baik	1	0	0	0	0
5	milenial	1	0	0	0	0
6	investasi	1	0	0	0	0
7	aplikasi	0	1	0	0	0
8	rapi	0	1	0	0	0
9	mudah	0	1	0	1	0
10	guna	0	1	0	1	0

Tabel 3.9 Lanjutan perhitungan TF

11	sebenarnya	0	0	1	0	0
12	kecewa	0	0	1	0	0
13	kembali	0	0	1	0	0
14	dana	0	0	1	0	0
15	lama	0	0	1	0	0
16	friendly	0	0	0	1	0
17	user	0	0	0	1	0
18	tampil	0	0	0	1	0
19	enak	0	0	0	1	0
20	cek	0	0	0	0	1
21	tidak	0	0	0	0	1
22	bisa	0	0	0	0	1
23	buka	0	0	0	0	1

3.1.5.2. Perhitungan DF, IDF dan W

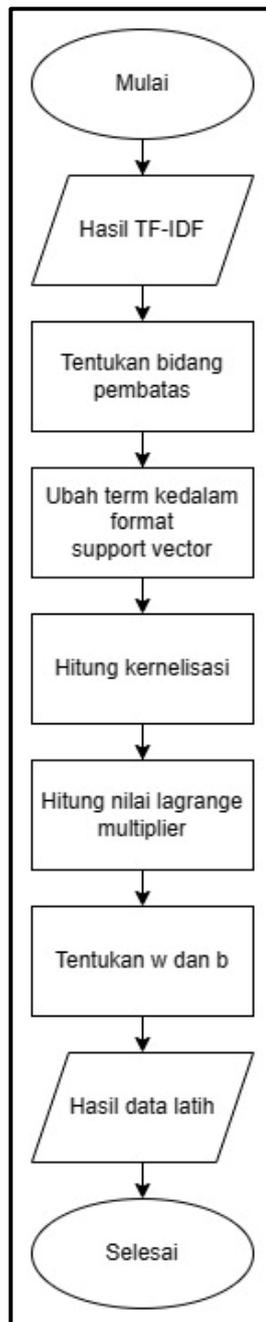
Proses perhitungan TF-IDF selanjutnya melibatkan *document frequency* (DF) yang merupakan banyaknya dokumen di mana suatu kata atau fitur muncul, kemudian *inverses document frequency*(IDF) merupakan hasil *inverse* dari DF dengan perhitungan $\log(D/DF)+1$ di mana D adalah jumlah keseluruhan dokumen dan W merupakan hasil dari TF-IDF yaitu bobot kata yang dihasilkan dari TF dikali dengan IDF. Berikut adalah tabel perhitungan TF-IDF seperti pada Tabel 3.10.

Tabel 3.10 Perhitungan DF, IDF dan TF-IDF

Indeks	Term	DF	IDF	W = TF * IDF				
				D1	D2	D3	D4	D5
0	cocok	1	1,698	1,698	0	0	0	0
1	mula	2	1,398	1,398	0	0	1,398	0
2	rekomendasi	1	1,698	1,698	0	0	0	0
3	beri	1	1,698	1,698	0	0	0	0
4	baik	1	1,698	1,698	0	0	0	0
5	milenial	1	1,698	1,698	0	0	0	0
6	investasi	1	1,698	1,698	0	0	0	0
7	aplikasi	1	1,698	0	1,698	0	0	0
8	rapi	1	1,698	0	1,698	0	0	0
9	mudah	2	1,397	0	1,398	0	1,398	0
10	guna	2	1,397	0	1,398	0	1,398	0
11	sebenarnya	1	1,698	0	0	1,698	0	0
12	kecewa	1	1,698	0	0	1,698	0	0
13	kembali	1	1,698	0	0	1,698	0	0
14	dana	1	1,698	0	0	1,698	0	0
15	lama	1	1,698	0	0	1,698	0	0
16	friendly	1	1,698	0	0	0	1,698	0
17	user	1	1,698	0	0	0	1,698	0
18	tampil	1	1,698	0	0	0	1,698	0
19	enak	1	1,698	0	0	0	1,698	0
20	cek	1	1,698	0	0	0	0	1,698
21	tidak	1	1,698	0	0	0	0	1,698

3.1.6. Tahap Pelatihan SVM

Pada tahap ini akan menggambarkan proses pelatihan *Support Vector Machine*. Hasil dari pembobotan TF-IDF yang telah dihasilkan sebelumnya akan menjadi data latih pada tahap ini. Tahap ini dilakukan dengan beberapa proses diantaranya adalah mendapat input dari hasil pembobotan kata TF-IDF, menentukan bidang pembatas, mengubah ke dalam format *support vector*, menghitung kernelisasi, menghitung nilai *lagrange multiplier*, menghitung nilai w dan b kemudian akan didapatkan hasil keluaran berupa hasil data latih. Berikut merupakan *flowchart* tahap pelatihan seperti Gambar 3.12.



Gambar 3.12 Diagram *flowchart* pelatihan SVM

3.1.6.1. Tentukan bidang pembatas

Menentukan bidang pembatas dengan menggunakan persamaan seperti di bawah misal dilakukan penentuan bidang pembatas terhadap data 1 maka akan didapatkan hasil seperti berikut:

$$D1: 1.698w_1 + 1.398w_2 + 1.698w_3 + 1.698w_4 + 1.698w_5 + 1.698w_6 + 1.698w_7 + b \leq 1$$

$$D5: -1.698w_1 - 1.698w_2 - 1.698w_3 - 1.698w_4 - b \geq 1$$

3.1.6.2. Ubah format data latih ke dalam format *support vector*

Tahap ini dilakukan untuk mengubah dari hasil bidang pembatas ke dalam format *support vector*. Pada tahap ini akan dilakukan pelabelan terhadap data latih. Untuk data sentimen positif diberi label 1 dan data sentimen negatif diberi label -1, seperti pada Tabel 3.11.

Tabel 3.11 Format data *support vector*

Term	D1(x1)	D2(x2)	D3(x3)	D4(x4)	D5(x5)
cocok	1,698	0	0	0	0
mula	1,398	0	0	1,398	0
rekomendasi	1,698	0	0	0	0
beri	1,698	0	0	0	0
baik	1,698	0	0	0	0
milennial	1,698	0	0	0	0
investasi	1,698	0	0	0	0
aplikasi	0	1,698	0	0	0
rapi	0	1,698	0	0	0
mudah	0	1,398	0	1,398	0
guna	0	1,398	0	1,398	0
sungguh	0	0	1,698	0	0
kecewa	0	0	1,698	0	0
kembali	0	0	1,698	0	0
dana	0	0	1,698	0	0
lama	0	0	1,698	0	0
friendly	0	0	0	1,698	0
user	0	0	0	1,698	0
tampil	0	0	0	1,698	0
enak	0	0	0	1,698	0
cek	0	0	0	0	1,698
tidak	0	0	0	0	1,698
bisa	0	0	0	0	1,698
buka	0	0	0	0	1,698
y (kelas)	1	1	-1	1	-1

3.1.6.3. Hitung kernelisasi

Kernelisasi menggunakan fungsi kernel *Radial Basis Function* (RBF). Misal dilakukan perhitungan untuk x_i , $i=1$ hingga 3 dan $\gamma = 1$.

$$K(x_1, x) = \exp \left(-1\sqrt{(1.698 - 1.698)^2 + (1.398 - 1.398)^2 + (1.698 - 1.698)^2} \right. \\ \left. + (1.698 - 1.698)^2 + (1.698 - 1.698)^2 + (1.698 - 1.698)^2 + 2 \right) = 1$$

$$K(x_1, x) = \exp \left(-1\sqrt{(0 - 1.698)^2 + (0 - 1.698)^2 + (0 - 1.398)^2 + (0 - 1.398)^2} \right) \\ = 0.04458$$

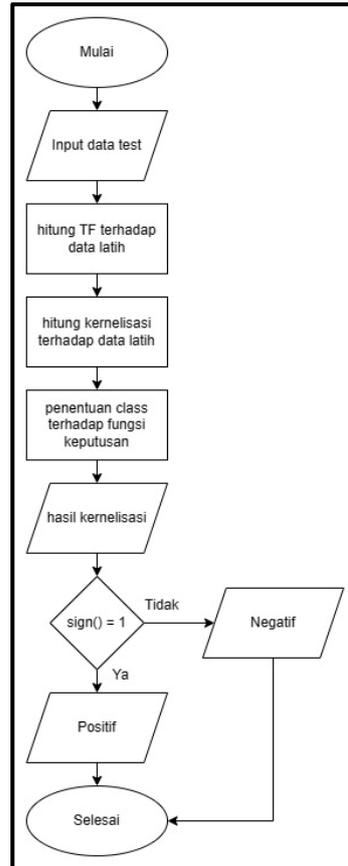
$$K(x_1, x) = \exp \left(-1\sqrt{(0 - 1.698)^2 + (0 - 1.698)^2 + (0 - 1.698)^2 + (0 - 1.698)^2} \right) \\ = 0.02244$$

Setelah dilakukan perhitungan terhadap semua data, maka akan didapatkan sebuah matriks $x_i x_j^T$ sebagai berikut:

$$x_i x_j^T = \begin{bmatrix} 1 & 0.01242 & 0.01242 & 0.01242 & 0.01242 \\ 0.04458 & 1 & 0.04458 & 0.04458 & 0.04458 \\ 0.02244 & 0.02244 & 1 & 0.02244 & 0.02244 \\ 0.03351 & 0.03351 & 0.03351 & 1 & 0.03351 \\ 0.03351 & 0.03351 & 0.03351 & 0.03351 & 1 \end{bmatrix}$$

3.1.7. Tahap Pengujian SVM

Tahap pengujian akan menguji model yang telah dihasilkan pada tahap pelatihan sebelumnya. Misalkan terdapat suatu ulasan aplikasi Bibit c = “Bibit mantap *deh*, metode pembayaran banyak tinggal pilih yang mana. Eksekusi ordernya, laporan transaksi cepat banget. Aplikasi ini *rekomendasi* banget *deh* pokoknya”, data tersebut akan dilakukan *pre-processing* sehingga menjadi x = “bibit mantap metode bayar banyak laporan cepat aplikasi rekomendasi”. Berikut merupakan *flowchart* dari tahap pengujian seperti pada gambar 3.13.



Gambar 3.13 Diagram *flowchart* pengujian SVM

3.1.8. Pengujian

Pada bagian ini akan dibuat suatu rancangan pengujian. Rancangan pengujian dilakukan menggunakan *dataset* yang telah didapatkan dengan perbandingan 70% sebagai data latih dan 30% sebagai data uji. Parameter pengujian pada penelitian ini adalah algoritma SVM tanpa optimasi parameter. Metode pengujian yang akan digunakan adalah *confusion matrix*.

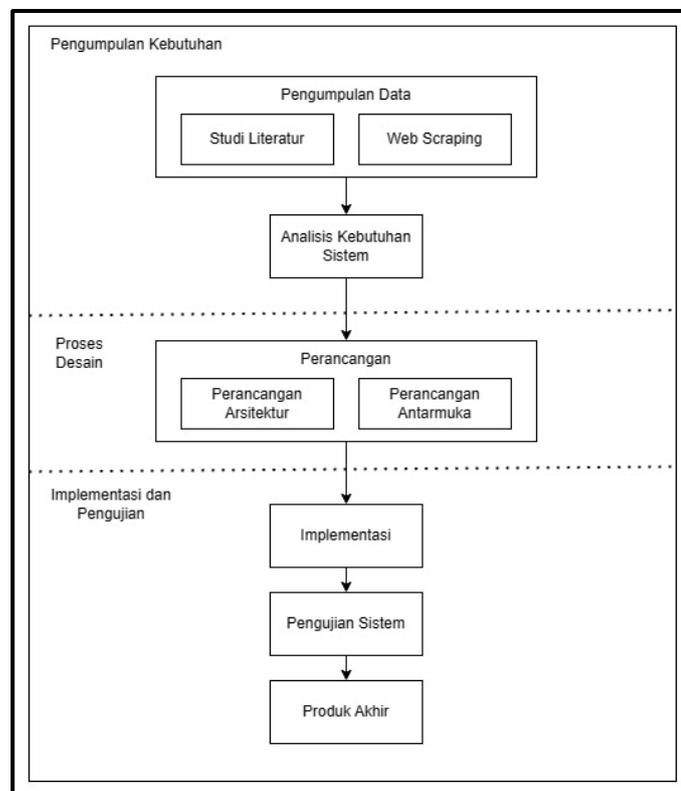
Pengujian *confusion matrix* bertujuan untuk mengukur tingkat akurasi, presisi dan *recall* dari suatu model. *Confusion matrix* dilakukan dengan menghitung nilai *true positive*, *false positive*, *true negative* dan *false negative*, kemudian nilai tersebut dapat digunakan untuk mengukur tingkat akurasi, presisi dan *recall*. Rancangan pengujian *confusion matrix* dapat dilihat pada tabel 3.12.

Tabel 3.12 Tabel rancangan *Confusion Matrix*

		Label Hasil Klasifikasi Sentimen	
		Positif	Negatif
Label Asli	Positif		
	Negatif		

3.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode pengembangan sistem *waterfall*. Metode pengembangan sistem *waterfall* dilakukan melalui tahap pengumpulan kebutuhan (*requirement definition*), perancangan (*system and software design*), implementasi dan pengujian (*implementation and testing*). Pengumpulan kebutuhan dilakukan untuk memenuhi kebutuhan yang akan digunakan dalam pengembangan sistem yang dilakukan. Kemudian tahap perancangan dilakukan untuk merancang arsitektur serta antarmuka dari sistem. Kemudian tahap implementasi dan pengujian akan dilakukan pada BAB selanjutnya. Metode pengembangan sistem yang dilakukan pada penelitian ini dapat dilihat seperti gambar 3.14.



Gambar 3.14 Diagram metode pengembangan sistem

3.2.1. Pengumpulan Kebutuhan

Pada bagian ini akan membahas mengenai kebutuhan-kebutuhan baik berupa data maupun informasi untuk dianalisis. Pengumpulan kebutuhan terdiri dari beberapa tahapan yaitu pengumpulan data, analisis data dan analisis kebutuhan sistem.

3.2.1.1. Pengumpulan data

Data merupakan salah satu komponen terpenting dalam penelitian. Penelitian memerlukan data untuk diolah sehingga menjadi sebuah informasi yang dapat mendukung

penelitian yang akan dilakukan. Pengumpulan data yang dilakukan dalam penelitian ini adalah menggunakan metode *web scraping*.

3.2.1.2. Analisis kebutuhan sistem

Pada langkah selanjutnya akan dilakukan analisis hal-hal yang berhubungan dengan kebutuhan-kebutuhan untuk mengembangkan sistem pada penelitian ini. Analisis ini bertujuan untuk mengidentifikasi kebutuhan dalam suatu penelitian. Analisis kebutuhan sistem terbagi menjadi dua bagian yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

a. Kebutuhan fungsional

Kebutuhan fungsional merupakan kebutuhan yang berkaitan langsung dengan sistem sehingga menggambarkan proses-proses yang akan diterapkan pada sistem ini. Berikut adalah kebutuhan fungsional dalam pengembangan sistem ini:

1. Sistem dapat melakukan proses *pre-processing* terhadap *dataset*.
2. Sistem dapat melakukan proses pembuatan model klasifikasi dengan metode *Support Vector Machine*.
3. Sistem dapat melakukan proses *training* dan *testing* menggunakan model klasifikasi *Support Vector Machine*.
4. Sistem dapat melakukan proses klasifikasi sentimen dan klasifikasi kategori.

b. Kebutuhan non-fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang meliputi properti perilaku yang dimiliki oleh sistem. Kebutuhan ini melibatkan beberapa perangkat yang mendukung sistem, yaitu kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*).

1. Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan untuk mengembangkan sistem pada penelitian ini dengan spesifikasi yang dapat dilihat pada Tabel 3.13.

Tabel 3.13 Kebutuhan perangkat keras

No	Perangkat Keras	Keterangan
1	<i>Processor</i>	Intel Core i5-12400f
2	<i>RAM</i>	16GB DDR4
3	<i>Storage</i>	SSD 512GB
4	<i>Graphic Card</i>	NVIDIA RTX 3060
5	Perangkat <i>input</i> dan <i>output</i>	Monitor, <i>Keyboard</i> dan <i>Mouse</i>
6	Koneksi internet	Wifi

2. Perangkat Lunak (*Software*)

Perangkat lunak yang dibutuhkan untuk mengembangkan sistem pada penelitian ini dapat dilihat pada Tabel 3.14.

Tabel 3.14 Kebutuhan perangkat lunak

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows 10 Pro 64 bit
2	<i>python</i>	Bahasa Pemrograman
3	Visual Studio Code	<i>Code Editor</i>
4	Microsoft Edge	<i>Web Browser</i>
5	Adobe Photoshop	Perangkat lunak untuk desain diagram dan rancangan antarmuka

3.2.2. Perancangan

Dalam tahapan ini akan membahas mengenai perancangan sistem yang akan dibangun. Perancangan yang akan dilakukan dalam pengembangan sistem analisis sentimen aplikasi reksa dana Bibit berbahasa Indonesia antara lain adalah perancangan arsitektur sistem, perancangan proses, perancangan antarmuka dan perancangan pengujian.

3.2.2.1. Perancangan arsitektur

Arsitektur sistem yang akan dibuat terdiri dari pengguna, model SVM dan sistem akan melakukan proses menampilkan data dan tabel *confusion matrix* serta nilai akurasi presisi dan *recall*. Pengguna adalah orang yang dapat menggunakan sistem ini. Sistem menggunakan data berupa teks ulasan aplikasi Bibit yang diambil secara acak dari halaman *web google play store* melalui proses *scraping*, data yang telah diolah pada tahap *text pre-processing* akan dilakukan pembobotan TF-IDF sehingga menghasilkan suatu *dataset* yang siap digunakan. Sistem akan melakukan proses identifikasi berdasarkan model *Support Vector Machine* yang telah dibuat sebelumnya.

3.2.2.2. Perancangan antarmuka

Bagian ini akan memuat hal yang berkaitan dengan perancangan antarmuka sistem. Perancangan antarmuka sistem penelitian ini meliputi struktur menu dan *user interface* (UI). Perancangan struktur menu dilakukan untuk menggambarkan menu-menu beserta sub-menu yang terdapat pada sistem dan perancangan UI dilakukan untuk menggambarkan tampilan antarmuka sistem.

Perancangan *user interface* merupakan model mekanisme komunikasi antara pengguna dengan sistem. Antarmuka atau *user interface* dapat menerima informasi dan memberikan informasi pada pengguna dalam bentuk tampilan yang interaktif. Berikut merupakan rancangan *user interface* pada sistem:

1. Rancangan Halaman Cek Ulasan

Halaman ulasan akan muncul pertama kali saat diakses. Halaman ini akan menampilkan panel input untuk memasukkan ulasan aplikasi Bibit. Setelah ulasan diproses maka akan menampilkan hasil analisis sentimen dari ulasan tersebut berupa sentimen positif ataupun negatif. Rancangan dari halaman ulasan dapat dilihat pada gambar 3.15.

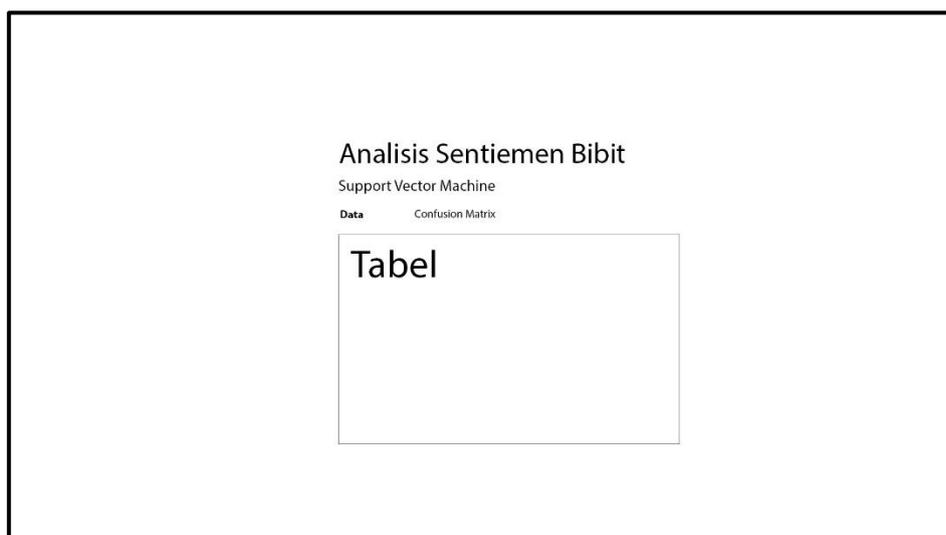


The image shows a wireframe for a web page titled "Analisis Sentimen Bibit". At the top, it says "Support Vector Machine". Below that, there are three tabs: "Cek Ulasan" (which is active), "Data", and "Confusion Matrix". Under the "Cek Ulasan" tab, there is a text input field labeled "Cek Ulasan Positif atau Negatif" and a "Cek" button. Below the input field, there is a label "Hasil text pre-processing dari ulasan yang dimasukkan" and a "Tabel" button. At the bottom, there is a line of text: "Ulasan yang anda masukkan berupa ulasan Positif".

Gambar 3.15 Rancangan halaman cek ulasan

2. Rancangan Halaman Data

Halaman data akan menampilkan semua *dataset* ulasan pengguna dan sentimen dari ulasan tersebut. Halaman ini juga menampilkan tabel *dataset* yang telah diproses *text pre-processing*, serta juga menampilkan tabel hasil proses *indexing* TF-IDF. Rancangan dari halaman data bisa dilihat pada gambar 3.16.

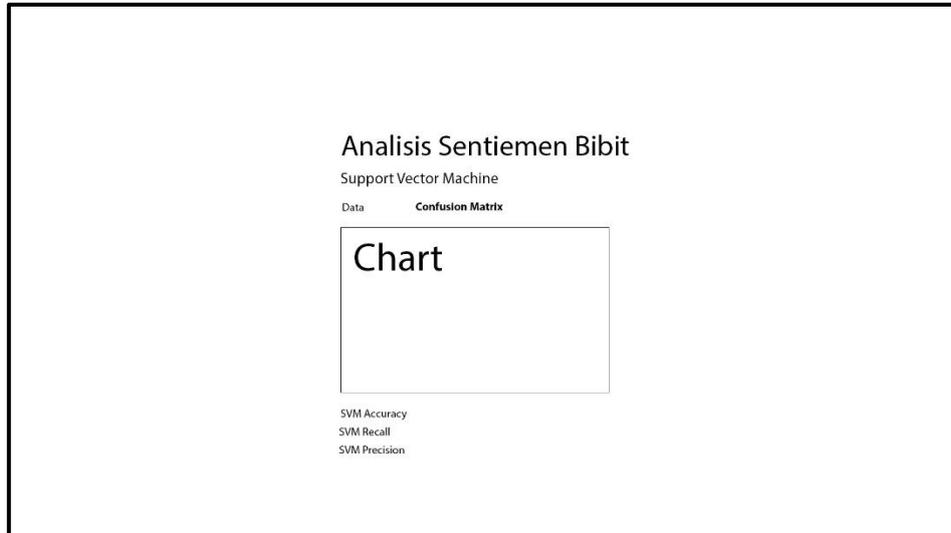


The image shows a wireframe for a web page titled "Analisis Sentimen Bibit". At the top, it says "Support Vector Machine". Below that, there are two tabs: "Data" (which is active) and "Confusion Matrix". Under the "Data" tab, there is a large rectangular area labeled "Tabel".

Gambar 3.16 Rancangan halaman data

3. Rancangan Halaman *Chart Confusion Matrix*

Halaman ini akan menampilkan *chart confusion matrix* dari model SVM yang telah dilakukan *training* dan *testing* sebelumnya. Pada halaman ini juga akan menampilkan tingkat akurasi, *recall*, dan presisi dari metode *Support Vector Machine*. Rancangan halaman ini dapat dilihat pada gambar 3.17.



Gambar 3.17 Rancangan halaman *confusion matrix*

BAB IV

HASIL, PENGUJIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

Pada bagian ini akan membahas mengenai hasil penelitian serta implementasi rancangan yang telah dibuat sebelumnya. Dari pembahasan ini, akan diketahui sistem ini telah mencapai tujuan yang diinginkan berdasarkan analisis dan perancangan yang telah dilakukan. Hasil penelitian akan dibagi menjadi 3 bagian yaitu pengumpulan data, implementasi analisis sentimen, dan implementasi sistem. Berikut penjelasan lebih lanjut.

4.1.1. Pengumpulan Data

Bagian ini akan membahas mengenai hasil pengumpulan data yang telah dilakukan sebelumnya, pengumpulan data dilakukan dengan cara *web scraping*. Data hasil *web scraping* akan digunakan untuk tahap pengujian.

Pengumpulan data menggunakan teknik *web scraping* dari halaman *web google play store* acak dengan menggunakan bahasa pemrograman *python* dengan bantuan *library google_play_scraper*. Proses *web scraping* dilakukan dengan mengambil setiap ulasan dan *score* yang diberikan oleh pengguna aplikasi Bibit. *Library google_play_scraper* akan melakukan *scraping* berdasarkan parameter yang dimasukkan yaitu id aplikasi, bahasa, negara asal, jenis sortir dan jumlah data yang akan diambil. Berikut modul *web scraping* pada modul program 4.1.

```
from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    app_id = 'com.bibit.bibitid',
    lang = 'id',
    country = 'id',
    sort = Sort.MOST_RELEVANT,
    count = 2000
)
```

Modul Program 4.1 *Web Scraping*

Dengan dijalankannya modul program 4.1 maka akan didapatkan data berupa ulasan dan *score* terhadap aplikasi Bibit yang akan disimpan ke dalam sebuah *file dataset* dengan format *csv*. Berikut merupakan hasil *web scraping* seperti pada tabel 4.1.

Tabel 4.1 Contoh data hasil *web scraping*

userName	content	score
imam nur cahyadi	Aplikasinya gk bisa di buka, udah tak restrat tetep aja gk mau masuk aplikasi bibit. yang ada malahan layar warna putih polos. Saya pakai telkomsel jaringannya full 4G+ tapi tetep aja gk mau ke aplikasinya. Mohon bantuanya	2
Syahrul kaka	Aplikasinya bagus untuk pemula , simple , tapi saat membuka aplikasi sangat lama sekali , dan saat membeli saham kenapa selalu saja tidak bisa , padahal kalau buat beli reksadan , obligasi dll bisa enak banget	4
Dana Dyaksa	Terima kasih bibit, aplikasi ini sangat membantu buat orang yang mau belajar investasi terutama buat yang masih sangat awam, selain baca baca artikel dan belajar dari pakar investasi aplikasi bibit ini memudahkan kita karena telah membimbing dengan sangat mudah dipahami, terimakasih yaa bibit.	5
Elga Rian Ardiyansa	Sudah saatnya bibit menghadirkan fitur dark mode di update aplikasi selanjutnya..agar kami bisa memilih antara pakai dark mode atau tidak... Seperti di stockbit	4
Zni20	kecewa sama bibit,, karena.. Pencairan Dana, atau Penjualan Sham terlalu memakan waktu hingga seminggu di saat pengguna sedang butuh uang,, - kecewanya di situ aja si, kalau soal aman 👍 tapi lemotnya masuk ke rekening itu-__	2

4.1.2. Pelabelan Data

Pada bagian ini akan dilakukan pelabelan terhadap data yang telah didapatkan dari tahap sebelumnya. Pada tahap sebelumnya didapatkan data ulasan aplikasi Bibit melalui proses *web scraping*. Data dilabeli secara manual dengan cara pengecekan terhadap komentar yang diberikan pada ulasan aplikasi Bibit. *Score* yang diberikan oleh pengguna pada halaman *web google play store* dijadikan pedoman tambahan dalam melakukan pelabelan data. Berikut merupakan hasil pelabelan yang telah dilakukan seperti pada tabel 4.2 dan tabel 4.3.

Tabel 4.2 Contoh data hasil pelabelan

User Name	Komentar	Sentimen
imam nur cahyadi	Aplikasinya gk bisa di buka, udah tak restrat tetep aja gk mau masuk aplikasi bibit. yang ada malahan layar warna putih polos. Saya pakai telkomsel jaringannya full 4G+ tapi tetep aja gk mau ke aplikasinya. Mohon bantuanya	Negatif
Syahrul kaka	Aplikasinya bagus untuk pemula , simple , tapi saat membuka aplikasi sangat lama sekali , dan saat membeli saham kenapa selalu saja tidak bisa , padahal kalau buat beli reksadan , obligasi dll bisa enak banget	Positif
Dana Dyaksa	Terima kasih bibit, aplikasi ini sangat membantu buat orang yang mau belajar investasi terutama buat yang masih sangat awam, selain baca baca artikel dan belajar dari pakar investasi aplikasi bibit ini memudahkan kita karena telah membimbing dengan sangat mudah dipahami, terimakasih yaa bibit.	Positif
Elga Rian Ardiyansa	Sudah saatnya bibit menghadirkan fitur dark mode di update aplikasi selanjutnya..agar kami bisa memilih antara pakai dark mode atau tidak... Seperti di stockbit	Positif

Tabel 4.3 Lanjutan contoh data hasil pelabelan

Zni20	kecewa sama bibit,, karena.. Pencairan Dana, atau Penjualan Sham terlalu memakan waktu hingga seminggu di saat pengguna sedang butuh uang,, - kecewanya di situ aja si, kalau soal aman 👍 tapi lemotnya masuk ke rekening itu-__	Negatif
-------	--	---------

4.1.3. Implementasi Analisis Sentimen Aplikasi Bibit

Implementasi analisis sentimen aplikasi investasi reksa dana Bibit yaitu hasil penelitian yang berkaitan dengan metode yang digunakan sesuai dengan tahapan dari identifikasi analisis sentimen pada ulasan aplikasi Bibit. Berikut merupakan proses identifikasi ulasan aplikasi Bibit :

4.1.3.1. Modul *text pre-processing*

Pada tahap ini, data ulasan akan dilakukan proses *text pre-processing* yang bertujuan untuk membersihkan data hingga siap digunakan. Data yang digunakan yaitu data hasil dari proses *web scraping*. Proses *text pre-processing* dilakukan melalui beberapa tahap. Berikut merupakan modul-modul yang digunakan pada proses *text pre-processing*.

a. Modul *cleansing*

Modul *cleansing* berfungsi untuk menghapus semua karakter selain huruf dan mengganti kata berulang dengan kata dasar. Implementasi dari proses *cleansing* menggunakan *regular expression* yang telah disediakan pada bahasa pemrograman *python*. Modul *cleansing* dapat dilihat pada modul program 4.2.

```
#proses cleansing
cleansing = []
for x in df['Komentar']:
    temporary = word_tokenize(str(x))
    temporary = [word for word in temporary if word.isalpha()]
    cleansing.append(temporary)

cleansing
```

Modul Program 4.2 Modul *cleansing*

Dengan dijalankannya modul program 4.2 maka akan didapatkan hasil *text pre-processing* tahap *cleansing* seperti pada tabel 4.4 dan tabel 4.5.

Tabel 4.4 Contoh data hasil *cleansing*

Komentar
Aplikasinya gk bisa di buka udah tak restrat tetep aja gk mau masuk aplikasi bibit yang ada malahan layar warna putih polos Saya pakai telkomsel jaringannya full 4G tapi tetep aja gk mau ke aplikasinya Mohon bantuanya
Aplikasinya bagus untuk pemula simple tapi saat membuka aplikasi sangat lama sekali dan saat membeli saham kenapa selalu saja tidak bisa padahal kalau buat beli reksadan obligasi dll bisa enak banget

Tabel 4.5 Lanjutan contoh data hasil *cleansing*

Terima kasih bibit aplikasi ini sangat membantu buat orang yang mau belajar investasi terutama buat yang masih sangat awam selain baca artikel dan belajar dari pakar investasi aplikasi bibit ini memudahkan kita karena telah membimbing dengan sangat mudah dipahami terimakasih yaa bibit
Sudah saatnya bibit menghadirkan fitur dark mode di update aplikasi selanjutnya agar kami bisa memilih antara pakai dark mode atau tidak Seperti di stockbit
kecewa sama bibit karena Pencairan Dana atau Penjualan Sham terlalu memakan waktu hingga seminggu di saat pengguna sedang butuh uang kecewanya di situ aja si kalau soal aman tapi lemotnya masuk ke rekening itu

b. Modul *word conversion*

Modul *word conversion* merupakan tahapan pada *text pre-processing* untuk menyeragamkan penggunaan kata. Implementasi modul ini adalah mengubah kata tidak baku menjadi kata baku. Kamus kata baku dan tidak baku didapatkan dari Github. Modul *word conversion* dapat dilihat pada modul program 4.3.

```

alay_dict_map = dict(zip(alay_dict['original'],
alay_dict['replacement']))
def normalize_alay(text):
    return ' '.join([alay_dict_map[word] if word in
alay_dict_map else word for word in text.split(' ')])

#proses normalisasi
def preprocessnormalisasi(data):
    text = normalize_alay(data)
    return text

#data preprocessing normalisasi
df['Komentar'] = df['Komentar'].apply(preprocessnormalisasi)
df = df[['Komentar', 'Sentimen']]

```

Modul Program 4.3 Modul *word conversion*

Dengan dijalankannya modul program 4.3 maka akan didapatkan hasil *text pre-processing* tahap *word conversion* seperti pada tabel 4.6 dan tabel 4.7.

Tabel 4.6 Contoh data hasil *word conversion*

Komentar
Aplikasinya tidak bisa di buka sudah tak restart tetap saja tidak mau masuk aplikasi bibit yang ada malahan layar warna putih polos Saya pakai telkomsel jaringannya full 4G tapi tetap saja tidak mau ke aplikasinya Mohon bantuannya
Aplikasinya bagus untuk pemula simpel tapi saat membuka aplikasi sangat lama sekali dan saat membeli saham kenapa selalu saja tidak bisa padahal kalau buat beli reksa dana obligasi bisa enak banget
Terima kasih bibit aplikasi ini sangat membantu buat orang yang mau belajar investasi terutama buat yang masih sangat awam selain baca artikel dan belajar dari pakar investasi aplikasi bibit ini memudahkan kita karena telah membimbing dengan sangat mudah dipahami terima kasih ya bibit
Sudah saatnya bibit menghadirkan fitur dark mode di update aplikasi selanjutnya agar kami bisa memilih antara pakai dark mode atau tidak Seperti di stockbit

Tabel 4.7 Lanjutan contoh data hasil *word conversion*

kecewa sama bibit karena Pencairan Dana atau Penjualan Saham terlalu memakan waktu hingga seminggu di saat pengguna sedang butuh uang kecewanya di situ saja si kalau soal aman tapi lemotnya masuk ke rekening itu

c. Modul *tokenizing*

Modul *tokenizing* merupakan tahapan pada *text pre-processing* yang digunakan untuk memotong data menjadi per kata. Data yang digunakan berasal dari hasil proses *word conversion*. Setiap kalimat pada data akan dipecah menjadi per kata. Implementasi modul *tokenizing* menggunakan fungsi *split()* yang telah disediakan oleh bahasa pemrograman *python*. Modul *tokenizing* dapat dilihat pada modul program 4.4.

```
#proses tokenizing
def tokenize_text(x):
    return x.split(' ')

tokeneddata=[]
for i,x in enumerate(cleansing):
    tokeneddata.append(" ".join(map(str, x)))
tfidf= TfidfVectorizer(tokenizer=tokenize_text,use_idf=True,
smooth_idf=True, sublinear_tf=False)
tfidf.fit_transform([tokeneddata[0]])
```

Modul Program 4.4 Modul *tokenizing*

Dengan dijalankannya modul program 4.4 maka akan didapatkan hasil *text pre-processing* tahap *tokenizing* seperti pada tabel 4.8.

Tabel 4.8 Contoh data hasil *tokenizing*

Komentar
['Aplikasinya', 'tidak', 'bisa', 'di', 'buka', 'sudah', 'tak', 'restart', 'tetap', 'saja', 'tidak', 'mau', 'masuk', 'aplikasi', 'bibit', 'yang', 'ada', 'malahan', 'layar', 'warna', 'putih', 'polos', 'Saya', 'pakai', 'telkomsel', 'jaringannya', 'full', '4G', 'tapi', 'tetap', 'saja', 'tidak', 'mau', 'ke', 'aplikasinya', 'Mohon', 'bantuannya']
['Aplikasinya', 'bagus', 'untuk', 'pemula', 'simpler', 'tapi', 'saat', 'membuka', 'aplikasi', 'sangat', 'lama', 'sekali', 'dan', 'saat', 'membeli', 'saham', 'kenapa', 'selalu', 'saja', 'tidak', 'bisa', 'padahal', 'kalau', 'buat', 'beli', 'reksa', 'dana', 'obligasi', 'bisa', 'enak', 'banget']
['Terima', 'kasih', 'bibit', 'aplikasi', 'ini', 'sangat', 'membantu', 'buat', 'orang', 'yang', 'mau', 'belajar', 'investasi', 'terutama', 'buat', 'yang', 'masih', 'sangat', 'awam', 'selain', 'baca', 'artikel', 'dan', 'belajar', 'dari', 'pakar', 'investasi', 'aplikasi', 'bibit', 'ini', 'memudahkan', 'kita', 'karena', 'telah', 'membimbing', 'dengan', 'sangat', 'mudah', 'dipahami', 'terima', 'kasih', 'ya', 'bibit']
['Sudah', 'saatnya', 'bibit', 'menghadirkan', 'fitur', 'dark', 'mode', 'di', 'update', 'aplikasi', 'selanjutnya', 'agar', 'kami', 'bisa', 'memilih', 'antara', 'pakai', 'dark', 'mode', 'atau', 'tidak', 'Seperti', 'di', 'stockbit']
['kecewa', 'sama', 'bibit', 'karena', 'Pencairan', 'Dana', 'atau', 'Penjualan', 'Saham', 'terlalu', 'memakan', 'waktu', 'hingga', 'seminggu', 'di', 'saat', 'pengguna', 'sedang', 'butuh', 'uang', 'kecewanya', 'di', 'situ', 'saja', 'si', 'kalau', 'soal', 'aman', 'tapi', 'lemotnya', 'masuk', 'ke', 'rekening', 'itu']

d. Modul *casefolding*

Modul *casefolding* berfungsi untuk mengubah semua karakter pada data judul menjadi huruf kecil (*lowercase*). Implementasi dari proses *casefolding* menggunakan fungsi *lower()* yang telah disediakan pada bahasa pemrograman *python*. Modul *casefolding* dapat dilihat pada modul program 4.5.

```
#proses casefolding
new_data['Komentar'] = new_data['Komentar'].str.lower()
```

Modul Program 4.5 Modul *casefolding*

Dengan dijalankannya modul program 4.5 maka akan didapatkan hasil *text pre-processing* tahap *casefolding* seperti pada tabel 4.9.

Tabel 4.9 Contoh data hasil *casefolding*

Komentar
['aplikasinya', 'tidak', 'bisa', 'di', 'buka', 'sudah', 'tak', 'restart', 'tetap', 'saja', 'tidak', 'mau', 'masuk', 'aplikasi', 'bibit', 'yang', 'ada', 'malahan', 'layar', 'warna', 'putih', 'polos', 'saya', 'pakai', 'telkomsel', 'jaringannya', 'full', '4g', 'tapi', 'tetap', 'saja', 'tidak', 'mau', 'ke', 'aplikasinya', 'mohon', 'bantuannya']
['aplikasinya', 'bagus', 'untuk', 'pemula', 'simplen', 'tapi', 'saat', 'membuka', 'aplikasi', 'sangat', 'lama', 'sekali', 'dan', 'saat', 'membeli', 'saham', 'kenapa', 'selalu', 'saja', 'tidak', 'bisa', 'padahal', 'kalau', 'buat', 'beli', 'reksa', 'dana', 'obligasi', 'bisa', 'enak', 'banget']
['terima', 'kasih', 'bibit', 'aplikasi', 'ini', 'sangat', 'membantu', 'buat', 'orang', 'yang', 'mau', 'belajar', 'investasi', 'terutama', 'buat', 'yang', 'masih', 'sangat', 'awam', 'selain', 'baca', 'artikel', 'dan', 'belajar', 'dari', 'pakar', 'investasi', 'aplikasi', 'bibit', 'ini', 'memudahkan', 'kita', 'karena', 'telah', 'membimbing', 'dengan', 'sangat', 'mudah', 'dipahami', 'terima', 'kasih', 'ya', 'bibit']
['sudah', 'saatnya', 'bibit', 'menghadirkan', 'fitur', 'dark', 'mode', 'di', 'update', 'aplikasi', 'selanjutnya', 'agar', 'kami', 'bisa', 'memilih', 'antara', 'pakai', 'dark', 'mode', 'atau', 'tidak', 'seperti', 'di', 'stockbit']
['kecewa', 'sama', 'bibit', 'karena', 'pencairan', 'dana', 'atau', 'penjualan', 'saham', 'terlalu', 'memakan', 'waktu', 'hingga', 'seminggu', 'di', 'saat', 'pengguna', 'sedang', 'butuh', 'uang', 'kecewanya', 'di', 'situ', 'saja', 'si', 'kalau', 'soal', 'aman', 'tapi', 'lemotnya', 'masuk', 'ke', 'rekening', 'itu']

e. Modul *stopword removal*

Modul *stopword removal* merupakan tahapan pada *text pre-processing* untuk menghapus kata yang kurang bermakna seperti kata hubung seperti 'dan', 'dengan', 'atau', dan sebagainya atau kata ganti seperti 'saya', 'kamu', 'dia' dan sebagainya. Implementasi modul *stopword removal* adalah dengan menghapus setiap kata pada masukan (input) yang didapatkan dari tahap sebelumnya yaitu tahap *casefolding*. Setiap kata akan dilakukan pengecekan terhadap kamus data kata *stopword*, apabila kata tersebut termasuk ke dalam *stopword* maka kata tersebut akan dihilangkan. Kamus data *stopword* didapatkan dengan menggunakan *library* NLTK yang tersedia untuk bahasa pemrograman *python*. Modul *stopword removal* dapat dilihat seperti pada modul program 4.6.

```
# proses stopwords
stop_words = set(stopwords.words('indonesian'))

new_data['Komentar'] = new_data['Komentar'].str.split('
').apply(lambda x: ' '.join(k for k in x if k not in
stop_words))
```

Modul Program 4.6 Modul *stopword removal*

Dengan dijalankannya modul program 4.6 maka akan didapatkan hasil *text pre-processing* tahap *stopword removal* seperti pada tabel 4.10.

Tabel 4.10 Contoh data hasil *stopword removal*

Komentar
['aplikasinya', 'tidak', 'buka', 'restart', 'tidak', 'masuk', 'aplikasi', 'bibit', 'layar', 'warna', 'putih', 'polos', 'pakai', 'telkomsel', 'jaringannya', 'full', 'tidak', 'aplikasinya', 'mohon', 'bantuannya']
['aplikasinya', 'bagus', 'pemula', 'simpler', 'membuka', 'aplikasi', 'membeli', 'saham', 'tidak', 'beli', 'reksa', 'dana', 'obligasi', 'anak', 'banget']
['terima', 'kasih', 'bibit', 'aplikasi', 'membantu', 'orang', 'belajar', 'investasi', 'awam', 'baca', 'artikel', 'belajar', 'pakar', 'investasi', 'aplikasi', 'bibit', 'memudahkan', 'membimbing', 'mudah', 'dipahami', 'terima', 'kasih', 'bibit']
['bibit', 'menghadirkan', 'fitur', 'dark', 'mode', 'update', 'aplikasi', 'memilih', 'pakai', 'dark', 'mode', 'tidak', 'stockbit']
['kecewa', 'bibit', 'pencairan', 'dana', 'penjualan', 'saham', 'memakan', 'seminggu', 'pengguna', 'butuh', 'uang', 'kecewanya', 'situ', 'aman', 'lemotnya', 'masuk', 'rekening']

f. Modul stemming

Modul *stemming* merupakan tahapan pada *text pre-processing* yang digunakan untuk mengubah kata menjadi kata dasar. Data yang digunakan berasal dari hasil proses *stopword removal*. Implementasi modul *stemming* menggunakan *library sastrawi* yang telah disediakan untuk bahasa pemrograman *python*. Modul *stemming* dapat dilihat pada modul program 4.7.

```
#proses stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()

words = []
for i, sentence in enumerate(new_data['Komentar']):
    output = stemmer.stem(sentence)
    words.append(output)

new = pd.DataFrame()
new['Komentar'] = pd.Series(words)

new_data['Komentar'] = new['Komentar']
```

Modul Program 4.7 Modul *stemming*

Dengan dijalankannya modul program 4.7 maka akan didapatkan hasil *text pre-processing* tahap *stemming* seperti pada tabel 4.11.

Tabel 4.11 Contoh data hasil *stemming*

Komentar
['aplikasi', 'tidak', 'buka', 'restart', 'tidak', 'masuk', 'aplikasi', 'bibit', 'layar', 'warna', 'putih', 'polos', 'pakai', 'telkomsel', 'jaring', 'full', 'tidak', 'aplikasi', 'mohon', 'bantu']
['aplikasi', 'bagus', 'mula', 'simpler', 'buka', 'aplikasi', 'beli', 'saham', 'tidak', 'beli', 'reksa', 'dana', 'obligasi', 'enak', 'banget']
['terima', 'kasih', 'bibit', 'aplikasi', 'bantu', 'orang', 'ajar', 'investasi', 'awam', 'baca', 'artikel', 'ajar', 'pakar', 'investasi', 'aplikasi', 'bibit', 'mudah', 'bimbing', 'mudah', 'paham', 'terima', 'kasih', 'bibit']
['bibit', 'hadir', 'fitur', 'dark', 'mode', 'update', 'aplikasi', 'pilih', 'pakai', 'dark', 'mode', 'tidak', 'stockbit']
['kecewa', 'bibit', 'cair', 'dana', 'jual', 'saham', 'makan', 'minggu', 'guna', 'butuh', 'uang', 'kecewa', 'situ', 'aman', 'lemot', 'masuk', 'rekening']

4.1.3.2. Modul pembobotan kata TF-IDF

Pada modul pembobotan kata TF-IDF ini akan memberi nilai atau bobot terhadap setiap kata atau fitur. Implementasi dari pembobotan kata TF-IDF dilakukan dengan cara mengalikan hasil *term frequency* (TF) dengan hasil *inverse document frequency* (IDF). Modul pembobotan kata TF-IDF dapat dilihat pada modul program 4.8.

```
#proses indexing tf-idf
tf_transformer = TfidfVectorizer()
tf_transformer.fit_transform(X_train)
X_train_tf = tf_transformer.transform(X_train)
X_test = tf_transformer.transform(X_test)
```

Modul Program 4.8 Modul pembobotan kata TF-IDF

Modul program 4.8 digunakan untuk menghitung bobot setiap kata dari semua *dataset*. Pada modul program 4.8 menggunakan fungsi *TfidfVectorizer* di mana fungsi tersebut digunakan untuk menghitung frekuensi kata pada setiap dokumen atau dapat disebut *term frequency* (TF). Selanjutnya adalah fungsi tersebut melakukan perhitungan nilai *inverse document frequency* (IDF) yang merupakan *inverse* dari nilai setiap dokumen yang memiliki *term* atau disebut dengan *document frequency* (DF). Terakhir melakukan perkalian hasil TF dengan IDF. Dengan dijalankannya modul program 4.8 maka akan didapatkan hasil pembobotan kata TF-IDF seperti pada tabel 4.12 dan tabel 4.13.

Tabel 4.12 Contoh hasil pembobotan kata TF-IDF

Indeks	Term	TF					DF	IDF	W = TF * IDF				
		D1	D2	D3	D4	D5			D1	D2	D3	D4	D5
0	aplikasi	2	2	2	1	0	7	1,397	1,397	0	0	0	1,397
1	tidak	2	1	0	1	0	4	1,096	2,193	1,096	0	1,096	0
2	buka	1	1	0	0	0	2	1,397	1,397	1,397	0	0	0
3	restart	1	0	0	0	0	1	1,698	1,698	0	0	0	0
4	masuk	1	0	0	0	0	1	1,698	1,698	0	0	0	0

Tabel 4.13 Lanjutan contoh hasil pembobotan kata TF-IDF

Indeks	Term	TF					DF	IDF	W = TF * IDF				
		D1	D2	D3	D4	D5			D1	D2	D3	D4	D5
5	bibit	1	0	3	1	1	6	0,920	0,920	0	2,762	0,920	0,920
6	layar	1	0	0	0	0	1	1,698	1,698	0	0	0	0
7	warna	1	0	0	0	0	1	1,698	1,698	0	0	0	0
8	putih	1	0	0	0	0	1	1,698	1,698	0	0	0	0
9	polos	1	0	0	0	0	1	1,698	1,698	0	0	0	0
10	pakai	1	0	0	0	0	1	1,698	1,698	0	0	0	0
11	telkomsel	1	0	0	0	0	1	1,698	1,698	0	0	0	0
12	jaring	1	0	0	0	0	1	1,698	1,698	0	0	0	0
13	full	1	0	0	0	0	1	1,698	1,698	0	0	0	0
14	mohon	1	0	0	0	0	1	1,698	1,698	0	0	0	0
15	bantu	1	0	1	0	0	2	1,397	1,397	0	1,397	0	0
16	bagus	0	1	0	0	0	1	1,698	0	1,698	0	0	0
17	mula	0	1	0	0	0	1	1,698	0	1,698	0	0	0
18	simpel	0	1	0	0	0	1	1,698	0	1,698	0	0	0
19	beli	0	2	0	0	0	2	1,397	0	2,795	0	0	0
20	saham	0	1	0	0	1	2	1,397	0	1,397	0	0	1,397
21	reksa	0	1	0	0	0	1	1,698	0	1,698	0	0	0
22	dana	0	1	0	0	1	2	1,397	0	1,397	0	0	1,397
23	obligasi	0	1	0	0	0	1	1,698	0	1,698	0	0	0
24	enak	0	1	0	0	0	1	1,698	0	1,698	0	0	0
25	banget	0	1	0	0	0	1	1,698	0	1,698	0	0	0
26	terima	0	0	2	0	0	2	1,397	0	0	2,795	0	0
27	kasih	0	0	2	0	0	2	1,397	0	0	2,795	0	0
28	orang	0	0	1	0	0	1	1,698	0	0	1,698	0	0
29	ajar	0	0	2	0	0	2	1,397	0	0	2,795	0	0
30	investasi	0	0	2	0	0	2	1,397	0	0	2,795	0	0
31	awam	0	0	1	0	0	1	1,698	0	0	1,698	0	0
32	baca	0	0	1	0	0	1	1,698	0	0	1,698	0	0
33	artikel	0	0	1	0	0	1	1,698	0	0	1,698	0	0
34	pakar	0	0	1	0	0	1	1,698	0	0	1,698	0	0
35	mudah	0	0	2	0	0	2	1,397	0	0	2,795	0	0
36	paham	0	0	1	0	0	1	1,698	0	0	1,698	0	0
37	bimbing	0	0	1	0	0	1	1,698	0	0	1,698	0	0
38	hadir	0	0	0	1	0	1	1,698	0	0	0	1,698	0
39	fitur	0	0	0	1	0	1	1,698	0	0	0	1,698	0
40	dark	0	0	0	2	0	2	1,397	0	0	0	2,795	0
41	mode	0	0	0	2	0	2	1,397	0	0	0	2,795	0
42	update	0	0	0	1	0	1	1,698	0	0	0	1,698	0
43	pilih	0	0	0	1	0	1	1,698	0	0	0	1,698	0
44	pakai	0	0	0	1	0	1	1,698	0	0	0	1,698	0
45	stockbit	0	0	0	1	0	1	1,698	0	0	0	1,698	0
46	kecewa	0	0	0	0	2	2	1,397	0	0	0	0	2,795
47	cair	0	0	0	0	1	1	1,698	0	0	0	0	1,698
48	jual	0	0	0	0	1	1	1,698	0	0	0	0	1,698
49	makan	0	0	0	0	1	1	1,698	0	0	0	0	1,698
50	minggu	0	0	0	0	1	1	1,698	0	0	0	0	1,698
51	guna	0	0	0	0	1	1	1,698	0	0	0	0	1,698
52	butuh	0	0	0	0	1	1	1,698	0	0	0	0	1,698
53	uang	0	0	0	0	1	1	1,698	0	0	0	0	1,698
54	situ	0	0	0	0	1	1	1,698	0	0	0	0	1,698
55	aman	0	0	0	0	1	1	1,698	0	0	0	0	1,698
56	masuk	0	0	0	0	1	1	1,698	0	0	0	0	1,698
57	rekening	0	0	0	0	1	1	1,698	0	0	0	0	1,698

4.1.3.3. Modul *support vector machine*

Pada tahap ini akan membahas mengenai modul dari algoritma *Support Vector Machine* (SVM). Proses ini menggunakan data dari hasil proses sebelumnya yaitu pembobotan kata TF-IDF. Penggunaan algoritma *Support Vector Machine* (SVM) pada penelitian ini terdiri dari tiga bagian yaitu proses pelatihan (*training*) dan proses pengujian (*testing*).

a. Modul *training*

Tahap ini merupakan proses untuk pembuatan model algoritma SVM untuk identifikasi ulasan aplikasi Bibit. Pembuatan model ini dilakukan dengan cara melatih sistem dengan contoh ulasan beserta labelnya sebagai bahan pembelajaran bagi model. Data input yang digunakan merupakan data dari hasil pembobotan kata TF-IDF yang kemudian data tersebut akan ditempatkan ke dalam bidang pembatas sesuai dengan kelas atau label dari masing-masing data, kemudian data akan dihitung menggunakan fungsi kernel RBF. Implementasi pembuatan model SVM dilakukan dengan menggunakan *library* dari *sklearn*. Proses *training* model dapat dilihat pada modul program 4.9.

```
#proses klasifikasi SVM
model_svm = SVC(C=1.0, kernel='rbf', gamma='scale',
probability=True, tol=1e-3, cache_size=1024)
model_svm.fit(X_train_tf, y_train)
```

Modul Program 4.9 Modul *training* SVM

Pada modul program *training* SVM digunakan parameter $C=1.0$ dengan $\text{gamma}='scale'$ pada kernel RBF dengan $\text{probability}=\text{true}$.

b. Modul *testing*

Setelah dilakukannya proses *training* maka selanjutnya dilakukan *testing* dengan model yang telah dibuat pada tahap sebelumnya. Model akan memprediksi data *test* kemudian akan menghitung total data yang diprediksi dengan jawaban benar dan total data yang diprediksi dengan jawaban salah. Kemudian akan dilakukan perhitungan nilai akurasi, *recall* dan presisi menggunakan *confusion matrix*. Modul testing dapat dilihat pada modul program 4.10.

```
pred = model_svm.predict(X_test)
print(classification_report(y_test, pred))
print("SVM Acuracy -->", accuracy_score(y_test, pred)*100)
print("SVM Recall -->", recall_score(y_test, pred, pos_label=1,
average='binary')*100)
print("SVM Precision -->", precision_score(y_test, pred,
pos_label=1, average='binary')*100)
```

Modul Program 4.10 Modul *testing* SVM

Dengan dijalankannya modul program 4.10 maka akan didapatkan hasil testing seperti pada Gambar 4.1.

```
Data Train= (1400,)
Data Test= (600,)
[[ 98  61]
 [ 32 409]]
```

Gambar 4.1 Hasil *testing* SVM

4.1.4. Implementasi Sistem

Implementasi sistem merupakan bagian yang menampilkan hasil dari rancangan sistem yang telah dibuat pada bab sebelumnya. Rancangan sistem berisi tampilan antarmuka beserta fungsi-fungsinya. Halaman antarmuka dibagi menjadi beberapa bagian, yaitu:

1. Halaman Cek Ulasan

Halaman cek ulasan adalah halaman pertama yang akan tampil pada sistem ini. Pada halaman ini pengguna dapat melakukan pengecekan jenis ulasan pada aplikasi Bibit bersifat positif maupun negatif. Tampilan dari halaman cek ulasan dapat dilihat pada gambar 4.2.



Gambar 4.2 Halaman cek ulasan

2. Halaman Data

Pada halaman data akan menampilkan tabel *dataset* yang digunakan untuk proses *training* dan *testing* pada sistem analisis sentimen ini. Selain itu pada halaman ini juga menampilkan tabel *dataset* yang telah dilakukan proses *text pre-processing*. Tabel terakhir pada halaman ini merupakan tabel yang menampilkan *dataset* yang telah dilakukan pembobotan TF-IDF. Tampilan dari halaman data dapat dilihat pada gambar 4.3.

Analisis Sentimen Bibit
Support Vector Machine

Cek Ulasan ✓ Data Data Confusion Matrix

Dataset

Dataset yang digunakan sebagai data latih dan data tes

0	User Name	Komentar
0	1 Mamen Sadewo	2 tahun bersama bibit alhamdulillah masih lancar
1	2 917947	sangat mudah di mengerti
2	3 024_Terra Bumi Swari	Kode referral ku "tbswari" lumayan loh dapat cashback 25.000 ... Sejaul
3	4 111 234	Bagi yg mau mendaftar bisa memakai kode referral : ukht11 untuk men
4	5 114dany ArifHanto	Aplikasinya sangat bagus dan mudah di pahami untuk pemula. kalau ad
5	6 21006_Dei Mulia Ningrum	sangatlah bermanfaat bisa menabung jangka panjang sekaligus berinv
6	7 3318021 Ramanda Pratama	Aplikasinya mantap, mudah kita memilih reksadanya sesuai dengan
7	8 A T	Saya coba beli obligasi pakai uang 600.000 dari pemerintah karena sisa
8	9 A.R.A Official	Aplikasi sangat bagus cocok buat yang masih sekolah, dan jangan lupa
9	10 ABANG Marcopolo	Saran saya bagi pengguna pemula, atau pengguna baru , jgn ragu atau

Data hasil Text Pre-processing

Unnamed: 0	Komentar
0	tahun sama bibit alhamdulillah masih lancar
1	sangat mudah di erti
2	kode referral ku tbswari lumayan lah dapat cashback jauh ini lama aku pakai bibit transaksi ny
3	bagi yang mau daftar bisa pakai kode referral untuk dapat cashback besar kapan lagi investasi
4	aplikasi sangat bagus dan mudah di paham untuk mula kalau yang bilang kok lama banget ma
5	sangat manfaat bisa tabung jangka panjang sekaligus investasi reksadana dengan rekomen
6	aplikasi mantap mudah kita pilih reksadanya sesuai dengan profil risiko kita dan ada rekom
7	saya coba beli obligasi pakai uang dari pemerintah karena sisa sudah gagal buat usaha tahun lalu
8	aplikasi sangat bagus cocok buat yang masih sekolah dan jangan lupa untuk yang pertama kal
9	saran saya bagi guna mula atau guna baru jgn ragu atau bimbang mulai saja dulu investasi dar

Tabel TF-IDF

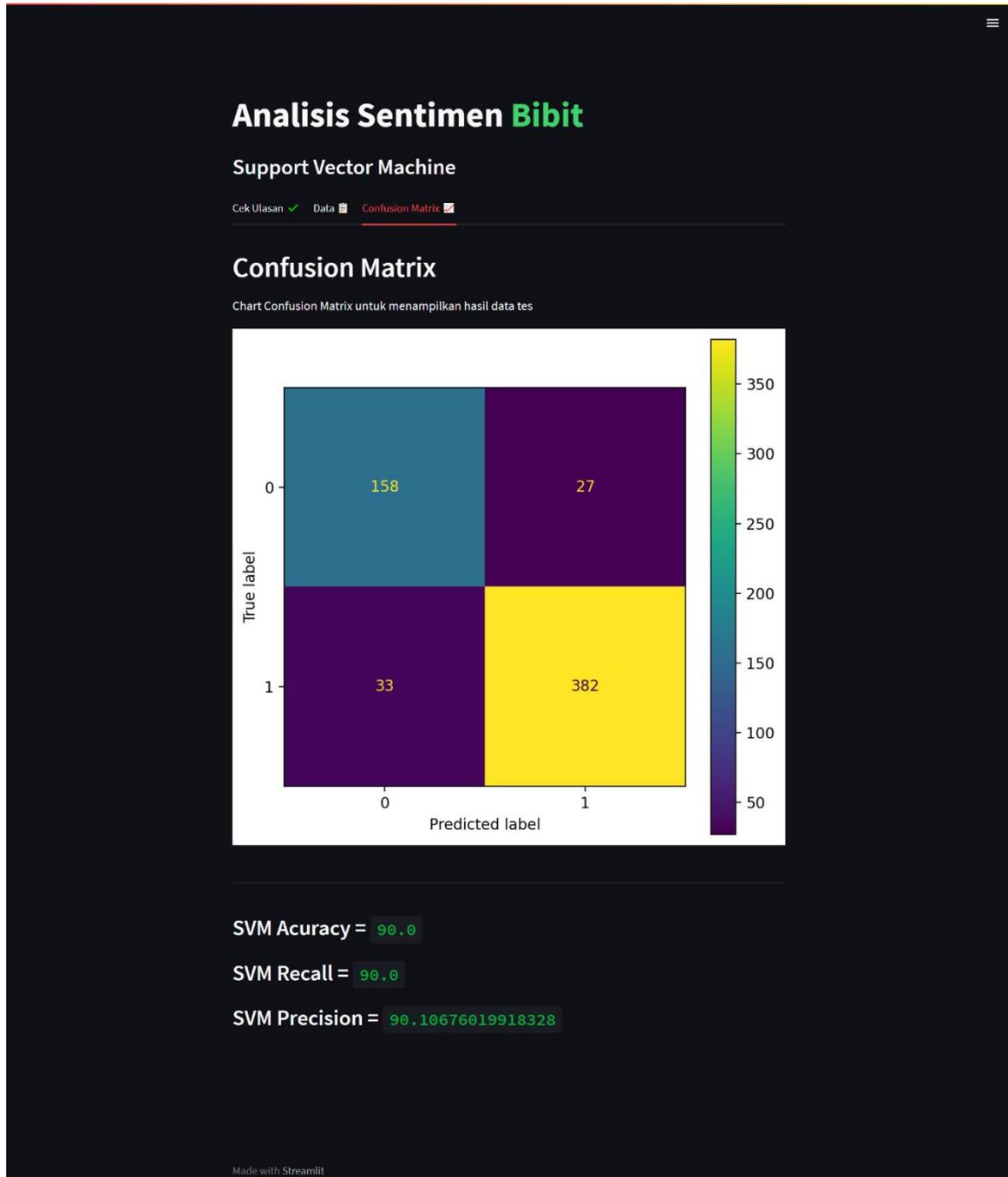
	abis	abu	acak	acc	accessibility	account	account	acung	ada	adakan	adalah	ac
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0

Made with Streamlit

Gambar 4.3 Halaman data

3. Halaman *Confusion Matrix*

Halaman *confusion matrix* akan menampilkan *chart confusion matrix* dari model SVM yang telah dilakukan *training* dan *testing* sebelumnya. Pada halaman ini juga akan menampilkan tingkat akurasi, *recall*, dan presisi dari metode *Support Vector Machine*. Tampilan dari halaman *confusion matrix* dapat dilihat pada gambar 4.4.



Gambar 4.4 Halaman *confusion matrix*

4.2. Pengujian

Bagian ini akan menjelaskan tentang hasil pengujian yang dilakukan terhadap hasil penelitian yang telah dibuat. Pengujian dilakukan dengan *dataset* yang telah didapatkan, *dataset* terdiri dari 2000 data. *Dataset* akan dibagi menjadi data latih dan data uji dengan pembagian data latih sejumlah 1400 atau sebesar 70% dari keseluruhan *dataset* dan data uji sejumlah 600 atau sebesar 30% dari keseluruhan *dataset*. Pengujian dilakukan dengan menggunakan *confusion matrix*.

Pengujian *confusion matrix* dilakukan untuk mengukur *performa* dari model *Support Vector Machine* (SVM). *Confusion matrix* dilakukan untuk mendapat nilai *True Positive* (TP), *False Positive* (FP), *True Negative* (TN) dan *False Negative* (FN), di mana nilai tersebut didapatkan dari hasil prediksi model *Support Vector Machine* (SVM) dengan label dari data yang sebenarnya. Data uji yang digunakan adalah sebanyak 600 data atau 30% dari keseluruhan data. Berikut merupakan contoh untuk memperoleh nilai TP, FP, TN dan FN dengan *confusion matrix* dapat dilihat pada tabel 4.14.

Tabel 4.14 Contoh *confusion matrix*

Data ke-	Label Sebenarnya	Hasil Prediksi	Nilai
1	Negatif	Negatif	TN
2	Positif	Positif	TP
3	Positif	Positif	TP
4	Positif	Negatif	FN
5	Negatif	Positif	TP

4.3. Pembahasan

Berdasarkan implementasi dan hasil pengujian yang dilakukan didapatkan hasil bahwa nilai akurasi, *recall*, dan presisi dari metode *Support Vector Machine* (SVM) dalam mengidentifikasi sentimen positif dan negatif dari ulasan yang diberikan terhadap aplikasi investasi reksa dana Bibit. Dari hasil pengujian algoritma SVM mendapat nilai akurasi sebesar 90%, nilai rata-rata *recall* sebesar 90%, dan nilai rata-rata presisi sebesar 90.1%.

Hasil pengujian menunjukkan bahwa penggunaan kernel RBF pada metode *Support Vector Machine* (SVM) dapat melakukan identifikasi ulasan mana yang merupakan ulasan positif maupun ulasan negatif dengan hasil yang baik.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil analisis, perancangan dan pembahasan yang dilakukan dapat diperoleh kesimpulan sebagai berikut:

1. Penggunaan kernel *Radial Basis Function* (RBF) pada metode *Support Vector Machine* (SVM) dapat menentukan jenis sentimen yang diberikan pada ulasan aplikasi investasi reksa dana Bibit.
2. Hasil pengujian yang dilakukan menunjukkan bahwa kernel *Radial Basis Function* (RBF) menghasilkan nilai akurasi sebesar 90%, nilai rata-rata *recall* sebesar 90%, dan nilai rata-rata presisi sebesar 90.1%

5.2. Saran

Adapun saran yang dapat digunakan untuk pengembangan penelitian lebih lanjut, antara lain sebagai berikut:

1. Proses pengumpulan data dapat dilakukan dengan cara menggunakan *app store* lainnya ataupun menggunakan media sosial.
2. Pada proses *text pre-processing* dilakukan *slang word conversion* yang lebih mendalam agar jumlah *noise* pada *dataset* berkurang.
3. Dapat ditambahkan pengecekan negasi pada analisis sentimen.
4. Dapat ditambahkan algoritma optimasi untuk metode *Support Vector Machine* (SVM) agar dapat meningkatkan nilai akurasi, *recall* dan presisi.
5. Dapat membandingkan dengan metode klasifikasi lainnya, sehingga dapat diketahui metode dengan performa yang lebih baik dalam melakukan analisis sentimen.

DAFTAR PUSTAKA

- Adhianto, D. (2020). *Investasi Reksa Dana Sebagai Alternatif Investasi Bagi Investor Pemula*. 4(1), 32–44. <https://doi.org/10.37339/jurnal>
- Ahyar, H., & Sukmana, D. J. (2020). *Buku Metode Penelitian Kualitatif & Kuantitatif*. <https://www.researchgate.net/publication/340021548>
- Anggraini, N., & Suroyo, H. (2019). Comparison of Sentiment Analysis against Digital Payment “T-cash and Go-pay” in Social Media Using Orange Data Mining. *Journal of Information Systems and Informatics*, 1(2), 152–163.
- Ardhani, B. A., Chamidah, N., & Saifudin, T. (2021). Sentiment Analysis Towards Kartu Prakerja Using Text Mining with Support Vector Machine and Radial Basis Function Kernel. *Journal of Information Systems Engineering and Business Intelligence*, 7(2), 119. <https://doi.org/10.20473/jisebi.7.2.119-128>
- Berita Pers Investor Pasar Modal Tembus 10 Juta*. (2022).
- Buntoro, G. A. (2017). Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. *Integer Journal Maret*, 1(1), 32–41.
- Cahyono, Y. (2017). Analisis Sentiment Pada Sosial Media Twitter Menggunakan Naïve Bayes Classifier Dengan Feature Selection Particle Swarm Optimization Dan Term Frequency. In *Jurnal Informatika Universitas Pamulang* (Vol. 2, pp. 14–19).
- Chandani, V., & Wahono, R. S. (2015). Komparasi Algoritma Klasifikasi Machine Learning Dan Feature Selection pada Analisis Sentimen Review Film. *Journal of Intelligent Systems*, 1(1), 55–59.
- Dhiyo, K., Wijaya, Y., & Karyawati, E. (2020). *The Effects of Different Kernels in SVM Sentiment Analysis on Mass Social Distancing*.
- Hasnain, M., Pasha, M. F., Ghani, I., Imran, M., Alzahrani, M. Y., & Budiarto, R. (2020). Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking. *IEEE Access*, 8, 90847–90861. <https://doi.org/10.1109/ACCESS.2020.2994222>
- Himawan, H., Putri, D. G., & Kaswidjanti, W. (2018). *Seminar Nasional Informatika 2018 (semnasIF 2018) UPN “Veteran” Yogyakarta*. <http://bsd.pendidikan.id>.
- Husada, H. C., & Paramita, A. S. (2021). Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM). *Teknika*, 10(1), 18–26. <https://doi.org/10.34148/TEKNIKA.V10I1.311>
- Indraswari, R., & Arifin, A. Z. (2017). RBF KERNEL OPTIMIZATION METHOD WITH PARTICLE SWARM OPTIMIZATION ON SVM USING THE ANALYSIS

- OF INPUT DATA'S MOVEMENT. *Jurnal Ilmu Komputer Dan Informasi*, 10(1), 36. <https://doi.org/10.21609/jiki.v10i1.410>
- Mahendrajaya, R., Buntoro, G. A., & Setyawan, M. B. (2019). ANALISIS SENTIMEN PENGGUNA GOPAY MENGGUNAKAN METODE LEXICON BASED DAN SUPPORT VECTOR MACHINE. *Jurnal Teknik Universitas Muhammadiyah Ponorogo*, 3(2), 52–63.
- Mase, J., Furqon, M. T., & Rahayudi, B. (2018). Penerapan Algoritme Support Vector Machine (SVM) Pada Pengklasifikasian Penyakit Kucing. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(10), 3648–3654. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2692>
- Meika Dinandra. (2020). PENINGKATAN BRAND EQUITY MELALU STRATEGI MARKETING COMMUNICATION APLIKASI INVESTASI REKSADANA “BIBIT.” *Jurnal Kalian Komunikasi*, 4(1), 29–40.
- Mejova, Y. (2009). *Sentiment Analysis: An Overview Comprehensive Exam Paper*. <http://www.pewinternet.org/Reports/2009/15-The-Internet-and-Civic-Engagement.aspx>
- Mufid, M. R., Basofi, A., Mawaddah, S., Khotimah, K., & Fuad, N. (2020). Risk diagnosis and mitigation system of covid-19 using expert system and web scraping. *IES 2020 - International Electronics Symposium: The Role of Autonomous and Intelligent Systems for Human Life and Comfort*, 577–583. <https://doi.org/10.1109/IES50839.2020.9231619>
- Muis, I. A., & Affandes, M. (2015). Penerapan Metode Support Vector Machine (SVM) Menggunakan Kernel Radial Basis Function (RBF) Pada Klasifikasi Tweet. *Jurnal Sains, Teknologi Dan Industri*, 12(2), 189–197. <http://ejournal.uin-suska.ac.id/index.php/sitekin>
- Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003a). Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika. *Kuliah Umum IlmuKomputer.Com*, 842–847.
- Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003b). *Support Vector Machine-Teori dan Aplikasinya dalam Bioinformatika 1*. <http://asnugroho.net>
- Nuzula, N. F., & Nurlaily, F. (2020). *Dasar-Dasar Manajemen Investasi*. Universitas Brawijaya Press.
- Prayitno, A., & Rifqi Ma'arif, M. (2018). Implementasi Web Scraping dan Text Mining untuk Akuisisi dan Kategorisasi Informasi Laman Web Tentang Hidroponik. *Indonesian Journal of Information Systems (IJIS)*, 1(1), 25–33.

- Rachman, F., & Purnami, S. W. (2012). Perbandingan Klasifikasi Tingkat Keganasan Breast Cancer Dengan Menggunakan Regresi Logistik Ordinal Dan Support Vector Machine (SVM). *JURNAL SAINS DAN SENI ITS*, 1(1).
- Sembiring, K. (2007). Penerapan teknik support vector machine untuk pendeteksian intrusi pada jaringan. *Institut Teknologi Bandung*.
- Utami, L. A. (2017). ANALISIS SENTIMEN OPINI PUBLIK BERITA KEBAKARAN HUTAN MELALUI KOMPARASI ALGORITMA SUPPORT VECTOR MACHINE DAN K-NEAREST NEIGHBOR BERBASIS PARTICLE SWARM OPTIMIZATION. *Jurnal Pilar Nusa Mandiri*, 13(1), 103–112.
- Wibawa, A. P., Guntur, M., Purnama, A., Fathony Akbar, M., & Dwiyanto, F. A. (2018). Metode-metode Klasifikasi. *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi*, 3(1).

LAMPIRAN

Tabel pembuktian hasil analisis sentimen

No.	Ulasan	Sentimen	Hasil Analisis Sentimen	Keterangan
1	Aplikasi bibit bagus sekali untuk orang yang baru belajar berinvestasi	Positif	Positif	Terbukti benar
2	Aplikasi bibit jelek sekali susah digunakan	Negatif	Negatif	Terbukti benar
3	Aplikasi tidak jelek	Positif	Negatif	Terbukti salah
4	Aplikasi tidak bagus	Negatif	Positif	Terbukti Salah
5	Pencairan dana lama sekali pakai aplikasi bibit	Negatif	Negatif	Terbukti benar
6	Aplikasi bibit sering sekali tidak bisa dibuka	Negatif	Negatif	Terbukti benar
7	Aplikasi bibit tidak tidak bagus	Positif	Negatif	Terbukti Salah
8	Aplikasi bibit tidak tidak jelek	Negatif	Negatif	Terbukti benar
9	Bibit cocok untuk menabung jangka panjang	Positif	Positif	Terbukti benar
10	aplikasi pembohong katanya bisa cepat bikin kaya ternyata tidak	Negatif	Negatif	Terbukti benar