

**SKRIPSI**

**ANALISIS SENTIMEN APLIKASI TWITTER TERHADAP  
TOPIK FINANSIAL MENGGUNAKAN ALGORITMA APRIORI  
DAN SUPPORT VECTOR MACHINES (SVM).**

**Oleh :**

**Nikodemus Ramot**

**065118044**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS PAKUAN  
BOGOR  
2023**

## **SKRIPSI**

# **ANALISIS SENTIMEN APLIKASI TWITTER TERHADAP TOPIK FINANSIAL MENGGUNAKAN ALGORITMA APRIORI DAN SUPPORT VECTOR MACHINES (SVM).**

Diajukan sebagai salah satu syarat memperoleh  
Gelar Sarjana Komputer Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam

**Oleh :**

**Nikodemus Ramot**

**065118044**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS PAKUAN  
BOGOR  
2023**



## HALAMAN PENGESAHAN

JUDUL : ANALISIS SENTIMEN APLIKASI TWITTER TERHADAP TOPIK FINANSIAL MENGGUNAKAN ALGORITMA APRIORI DAN SUPPORT VECTOR MACHINES (SVM).  
NAMA : Nikodemus Ramot  
NPM 065118044

Mengesahkan,

Pembimbing Pendamping  
FMIPA - UNPAK



(Siska Andriani, M.Kom.)

Pembimbing Utama  
FMIPA - UNPAK



(Arie Qur'ania, M.Kom.)

Mengetahui,

Ketua Program Studi Ilmu Komputer  
FMIPA - UNPAK



(Arie Qur'ania, M.Kom.)

Dekan  
FMIPA - UNPAK



(Asep Denih., S.Kom., M.Sc., Ph.D.)



## **PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI**

Dengan ini saya menyatakan bahwa :

Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah di publikasikan atau sudah pernah di pakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian – bagian di mana sumber informasinya dicantumkan dengan cara refrensi yang semestinya

Demikian pernyataan ini saya buat dengan sebenar – benarnya. Apabila kelas di kemudian hari terdapat gugatan, penulis bersedia di kenakan sanksi sesuai dengan peraturan yang berlaku

Bogor, 10 Juli 2023



Nikodemus Ramot Tanjaya

NPM. 065118044

## RIWAYAT HIDUP



Penulis dilahirkan di Bogor pada tanggal 15 bulan September tahun 2000 dari pasangan Bapak Laeder T.M.H.F dan Ibu Julianti sebagai anak tunggal. Penulis memulai pendidikan di sekolah dasar yang bertempat di SDK Satu Bakti di Bogor, lalu lulus dari SDK Ketapang di Cibubur di karenakan pindah ke SD Budi Murni Medan karena orang tua berpindah - pindah tempat tinggal.

Kemudian tahun 2015 masuk SMP Santa Lusia di Bekasi dan Penulis adalah Alumni dari SMA Mardiyuana Bogor. Pada tahun 2018 penulis meneruskan pendidikan ke Universitas Pakuan Bogor, Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam.

Selama di Universitas Pakuan, penulis pernah aktif di organisasi Keluarga Mahasiswa Katolik Pakuan (KMKP) periode 2018-2019 menjabat Anggota seksi logistik panitia natal sebagai anggota logistik panitia natal. Pada bulan Juli tahun 2023 penulis menyelesaikan penelitian dengan judul "Analisa Sentimen Aplikasi Twitter Terhadap Topik Finansial Menggunakan Algoritma Apriori dan Support Vector Machines".

## RINGKASAN

**Nikodemus Ramot Tanjaya 2023.** Analisa Sentimen Aplikasi Twitter Terhadap Topik Finansial Menggunakan Algoritma Apriori dan Support Vector Machines. Dibimbing oleh **Siska Andriani, M.Kom.,** dan **Arie Qur'ania, M.Kom.**

Penggunaan gawai pintar atau *smartphone* sangat meningkat tajam pada masa kini. Sehingga berefek pada kondisi perekonomian masyarakat, khususnya pada proses memperoleh informasi finansial, yang berdampak pada naik turunnya suatu saham, mata uang negara maupun mata uang crypto. Klasifikasi sentimen pengguna atau *user* pada saat menjelajahi aplikasi *Twitter* yang berbasis online, dalam penelitian ini peneliti menggunakan metode Apriori dan *Support Vector Machines*. Metode Apriori ini menerapkan salah satu algoritma klasik data mining. Algoritma apriori digunakan agar komputer dapat mempelajari aturan asosiasi, mencari pola hubungan antar satu atau lebih item dalam suatu dataset. Penelitian ini memiliki tujuan untuk mengklasifikasi data sentimen twitter pada tahun 2021 bulan Januari sampai tahun 2022 bulan November dengan menggunakan library pada bahasa pemrograman python untuk proses implementasi algoritma *support vector machines* dan algoritma apriori serta untuk mengetahui tingkat akurasi sentimen, dari beberapa kategori positif, negatif, netral, hampir positif dan hampir negatif. Data yang digunakan merupakan data yang diperoleh dari website kaggle yang sudah di ambil dari komentar twitter mengenai bitcoin crypto dalam bentuk *dataset* berbahasa Inggris berformat *csv*. Hasil dari proses penelitian ini menunjukkan bahwa algoritma support vector machines memiliki akurasi yang lebih baik sebesar 95% pada data uji, sedangkan untuk algoritma apriori sebesar 20% pada data ujinya.

## KATA PENGANTAR

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Kuasa atas segala rahmat dan karunia-Nya. Penulis telah menyelesaikan laporan skripsi yang berjudul “*Analisis Sentimen Aplikasi Twitter Terhadap Topik Finansial Menggunakan Algoritma Apriori Dan Support Vector Machines (SVM)*.”. Laporan Penelitian ini disusun sebagai salah satu syarat kelulusan di Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Pakuan Bogor. Dalam penyusunan Laporan Proposal Penelitian ini penulis banyak mendapat bantuan dari berbagai pihak, oleh sebab itu penulis ingin mengungkapkan rasa terimakasih kepada:

1. Arie Qur’ania, M.Kom., selaku pembimbing utama serta ketua program studi ilmu komputer yang selalu berkenan membantu dalam pengarahan dan bimbingan selama penyusunan laporan penelitian ini.
2. Siska Andriani, M.Kom., selaku pembimbing pendamping yang selalu berkenan membantu dalam pengarahan dan bimbingan selama penyusunan laporan penelitian ini.
3. Teristimewa kepada papa tercinta Laeder T.M.H.F dan mama tercinta Julianti, yang telah mengasuh dan membesarkan penulis, serta memberikan jalan terbaik pada penulis, membimbing dan mendoakan serta dukungan baik moril maupun materil.
4. Teman seperjuangan AB angkatan 2018, dan pacar saya Yoacлина.E.C. yang selalu memberikan dukungan dan motivasi kepada penulis selama ini.

Menyadari keterbatasan waktu dan kemampuan dalam penulisan laporan skripsi ini masih jauh dari sempurna. Oleh karena itu, segala kritik dan saran yang membangun akan diterima dengan senang hati. Mudah-mudahan Tuhan YME akan membalas semua kebaikan kepada semua pihak yang membantu. Akhir kata, semoga laporan ini dapat bermanfaat bagi kita semua. Amin.

Bogor, 10 Juli 2023

Nikodemus Ramot Tanjaya

065118044

## DAFTAR ISI

<b>HALAMAN PENGESAHAN</b> .....	i
<b>PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI</b> .....	ii
<b>RIWAYAT HIDUP</b> .....	iii
<b>RINGKASAN</b> .....	iv
<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vi
<b>DAFTAR GAMBAR</b> .....	ix
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR LAMPIRAN</b> .....	xi
<b>BAB I PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	2
<b>1.3 Ruang Lingkup</b> .....	3
<b>1.4 Tujuan Penelitian</b> .....	3
<b>1.5 Manfaat Penelitian</b> .....	3
<b>BAB II TINJAUAN PUSTAKA</b> .....	4
<b>2.1 Landasan Teori</b> .....	4
<b>2.1.1 Sistem Informasi</b> .....	4
<b>2.1.2 Twitter</b> .....	4
<b>2.1.3 Algoritma Support Vector Machines (SVM)</b> .....	4
<b>2.1.4 Algoritma Apriori</b> .....	5
<b>2.1.5 Analisis Sentimen</b> .....	5
<b>2.1.6 Text Mining</b> .....	5
<b>2.1.7 Python</b> .....	5
<b>2.1.8 Jupyter Notebook</b> .....	6
<b>2.1.8.1 NLTK</b> .....	6
<b>2.1.8.2 Numpy</b> .....	6
<b>2.1.8.3 Korpus</b> .....	6
<b>2.2 Tinjauan Pustaka</b> .....	6
<b>2.3 Tabel Perbandingan Penelitian Terdahulu</b> .....	6



2.4 Kerangka Teoritis .....	7
<b>BAB III METODE PENELITIAN .....</b>	<b>9</b>
3.1 Metode Penelitian.....	9
3.2 Literatur Review .....	9
3.3 Pengumpulan Data.....	10
3.4 Data Preprocessing .....	10
3.5 Pembobotan Term/Kata .....	10
3.6 Klasifikasi Data .....	11
3.6.1 Klasifikasi Support Vector Machine (SVM).....	12
3.6.2 Klasifikasi Apriori.....	14
3.6.2.1 <i>Support</i> .....	16
3.6.2.2 <i>Confidence</i> .....	16
3.7 Evaluasi .....	16
3.8 Analisis Hasil .....	16
<b>BAB IV RANCANGAN DAN IMPLEMENTASI.....</b>	<b>17</b>
4.1 Tahap Awal.....	17
4.1.1. Proses Pencarian Data .....	17
4.2 Tahap Pengembangan .....	18
4.2.1. Analisis Deskriptif.....	18
4.2.2. Sentimen.....	19
4.2.3. Data Cleaning dan Case Folding .....	19
4.2.4. Tokenizing.....	20
4.2.5. Stop Word Removal.....	21
4.2.6. Stemming .....	22
4.2.7. Pelabelan .....	23
4.2.8. Pembagian Data Latih dan Data Uji .....	23
4.2.9. Klasifikasi Support Vector Machine .....	24
4.2.10. Klasifikasi Asosiasi.....	24
4.3 Analisis Hasil dan Evaluasi .....	25
4.3.1. Visualisasi Data .....	25
4.3.2. Sentimen Positif.....	26
4.3.3. Sentimen Negatif .....	27
4.3.4. Sentimen Netral.....	28

4.3.5. Sentimen Hampir Positif .....	28
4.3.6. Sentimen Hampir Negatif .....	29
4.3.7. Evaluasi .....	30
<b>BAB V HASIL DAN PEMBAHASAN .....</b>	<b>31</b>
5.1 Hasil .....	31
5.2. Pembahasan .....	31
5.2.1 Klasifikasi SVM .....	31
5.2.2 Klasifikasi Asosiasi Rules dan Visualisasi .....	32
<b>BAB VI PENUTUP .....</b>	<b>34</b>
6.1 Kesimpulan .....	34
6.2 Saran .....	35
<b>DAFTAR PUSTAKA .....</b>	<b>36</b>
<b>LAMPIRAN .....</b>	<b>38</b>

## DAFTAR GAMBAR

<b>Gambar 1. Flowchart Kerangka Teoritis</b> .....	8
<b>Gambar 2. Diagram Alir Metode Penelitian</b> (Hendry Cipta Husada, 2021).....	9
<b>Gambar 3. Diagram Alir Data Preprocessing</b> (Hendry Cipta Husada, 2021) .....	10
<b>Gambar 4. Flowchart Algoritma SVM</b> (Husada, 2021) .....	11
<b>Gambar 5. Flowchart Algoritma Association Rule</b> (Husada, 2021) .....	11
<b>Gambar 6. Hyperplane Support Vector Machine</b> (Husada, 2021).....	12
<b>Gambar 7. Grafik SVM Linear</b> (Trivusi, 2022) .....	13
<b>Gambar 8. SVM Non-linear</b> (Trivusi, 2022) .....	14
<b>Gambar 9. Tampilan Dashboard Website Kaggle</b> (leoth9, 2022).....	17
<b>Gambar 10. Dataset Financial Tweet</b> (leoth9, 2022).....	18
<b>Gambar 11. Tampilan Data Klasifikasi Sentimen</b> .....	18
<b>Gambar 12. Proses Cleaning Data</b> .....	19
<b>Gambar 13. Proses label data sentimen</b> .....	23
<b>Gambar 14. Hasil Visualisasi Confusion Matrix Kernel Linear (80:20)</b> .....	24
<b>Gambar 15. Hasil Akurasi Data Latih (80:20) Klasifikasi Apriori</b> .....	25
<b>Gambar 16. Diagram Scatter Plot dan Garis Hyperplane Skenario Data (80:20)</b> .....	25
<b>Gambar 17. Grafik Pie Chart Sentimen Skenario 80:20</b> .....	26
<b>Gambar 18. Sebaran Kata Sentimen Positif (80:20)</b> .....	27
<b>Gambar 19. Sebaran Kata Sentimen Negatif (80:20)</b> .....	28
<b>Gambar 20. Sebaran Kata Sentimen Netral (80:20)</b> .....	28
<b>Gambar 21. Sebaran Kata Sentimen Hampir Positif (80:20)</b> .....	29
<b>Gambar 22. Sebaran Kata Sentimen Hampir Negatif (80:20)</b> .....	29
<b>Gambar 24. Hasil Penghitungan Akurasi Sentimen Klasifikasi SVM (80:20)</b> ..	32
<b>Gambar 25. Hasil Penghitungan Akurasi Emotikon Sentimen Klasifikasi SVM (80:20)</b> .....	32
<b>Gambar 26. Akurasi Data Latih Sentimen Klasifikasi Apriori (80:20)</b> .....	33
<b>Gambar 27. Akurasi Data Uji Sentimen Klasifikasi Apriori (80:20)</b> .....	33



## DAFTAR TABEL

<b>Tabel 1. Persamaan Setiap Kernel SVM (Husada, 2021)</b> .....	13
<b>Tabel 2. Tabel Kata Asosiasi Rule</b> .....	15
<b>Tabel 3. Proses cleaning data</b> .....	20
<b>Tabel 4. Proses tokenisasi data.</b> .....	20
<b>Tabel 5. Proses <i>Stop Word Removal</i></b> .....	21
<b>Tabel 6. Proses stemming data</b> .....	22
<b>Tabel 7. Pembagian Data</b> .....	23

## DAFTAR LAMPIRAN

Lampiran 1. Grafik Pie Chart Sentimen Skenario 75:25 .....	39
Lampiran 2. Grafik Pie Chart Sentimen Skenario 70:30 .....	39
Lampiran 3. Grafik Pie Chart Sentimen Skenario 65:35 .....	40
Lampiran 4. Hasil Visualisasi Confusion Matrix Kernel Linear (75:25) .....	41
Lampiran 5. Hasil Visualisasi Confusion Matrix Kernel Linear (70:30) .....	42
Lampiran 6. Hasil Visualisasi Confusion Matrix Kernel Linear (65:35) .....	43
Lampiran 7. Hasil Akurasi Data Uji (80:20) Klasifikasi Apriori.....	44
Lampiran 8. Hasil Akurasi Data Latih (75:25) Klasifikasi Apriori.....	44
Lampiran 9. Hasil Akurasi Data Uji (75:25) Klasifikasi Apriori.....	44
Lampiran 10. Hasil Akurasi Data Latih (70:30) Klasifikasi Apriori.....	45
Lampiran 11. Hasil Akurasi Data Uji (70:30) Klasifikasi Apriori.....	45
Lampiran 12. Hasil Akurasi Data Latih (65:35) Klasifikasi Apriori.....	46
Lampiran 13. Hasil Akurasi Data Uji (65:35) Klasifikasi Apriori.....	46
Lampiran 14. Diagram Scatter Plot dan Garis Hyperplane Skenario Data (75:25) .....	47
Lampiran 15. Diagram Scatter Plot dan Garis Hyperplane Skenario Data (70:30) .....	47
Lampiran 16. Diagram Scatter Plot dan Garis Hyperplane Skenario Data (65:35) .....	48
Lampiran 17. Sebaran Kata Wordcloud Sentimen Positif (80:20).....	48
Lampiran 18. Sebaran Kata Wordcloud Sentimen Negatif (80:20).....	49
Lampiran 19. Sebaran Kata Wordcloud Sentimen Netral (80:20) .....	49
Lampiran 20. Sebaran Kata Wordcloud Sentimen Hampir Positif (80:20).....	50
Lampiran 21. Sebaran Kata Wordcloud Sentimen Hampir Negatif (80:20) .....	50
Lampiran 22. Sebaran Kata Sentimen Positif (75:25).....	51
Lampiran 23. Sebaran Kata Word Cloud Sentimen Positif (75:25) .....	52
Lampiran 24. Sebaran Kata Sentimen Negatif (75:25) .....	52
Lampiran 25. Sebaran Kata Word Cloud Sentimen Negatif (75:25).....	53
Lampiran 26. Sebaran Kata Sentimen Netral (75:25).....	53
Lampiran 27. Sebaran Kata Wordcloud Sentimen Netral (75:25) .....	54
Lampiran 28. Sebaran Kata Sentimen Hampir Positif (75:25) .....	54
Lampiran 29. Sebaran Kata Wordcloud Sentimen Hampir Positif (75:25).....	55
Lampiran 30. Sebaran Kata Sentimen Hampir Negatif (75:25) .....	55
Lampiran 31. Sebaran Kata Wordcloud Sentimen Hampir Negatif (75:25) .....	56
Lampiran 32. Sebaran Kata Sentimen Positif (70:30).....	56
Lampiran 33. Sebaran Kata Wordcloud Sentimen Positif (70:30) .....	57
Lampiran 34. Sebaran Kata Sentimen Negatif (70:30) .....	57
Lampiran 35. Sebaran Kata Wordcloud Sentimen Negatif (70:30).....	58
Lampiran 36. Sebaran Kata Sentimen Netral (70:30).....	58
Lampiran 37. Sebaran Kata Wordcloud Sentimen Netral(70:30) .....	59
Lampiran 38. Sebaran Kata Sentimen Hampir Positif (70:30) .....	59

<b>Lampiran 39. Sebaran Kata Wordcloud Sentimen Hampir Positif (70:30) .....</b>	<b>60</b>
<b>Lampiran 40. Sebaran Kata Sentimen Hampir Negatif (70:30) .....</b>	<b>60</b>
<b>Lampiran 41. Sebaran Kata Wordcloud Sentimen Hampir Negatif (70:30).....</b>	<b>61</b>
<b>Lampiran 42. Sebaran Kata Sentimen Positif (65:35) .....</b>	<b>61</b>
<b>Lampiran 43. Sebaran Kata Wordcloud Sentimen Positif (65:35) .....</b>	<b>62</b>
<b>Lampiran 44. Sebaran Kata Sentimen Negatif (65:35).....</b>	<b>62</b>
<b>Lampiran 45. Sebaran Kata Wordcloud Sentimen Negatif (65:35) .....</b>	<b>63</b>
<b>Lampiran 46. Sebaran Kata Sentimen Netral (65:35) .....</b>	<b>63</b>
<b>Lampiran 47. Sebaran Kata Wordcloud Sentimen Netral (65:35) .....</b>	<b>64</b>
<b>Lampiran 48. Sebaran Kata Sentimen Hampir Positif (65:35).....</b>	<b>64</b>
<b>Lampiran 49. Sebaran Kata Sentimen Wordcloud Hampir Positif (65:35) .....</b>	<b>65</b>
<b>Lampiran 50. Sebaran Kata Sentimen Hampir Negatif (65:35) .....</b>	<b>65</b>
<b>Lampiran 51. Sebaran Kata Sentimen Wordcloud Hampir Negatif (65:35) .....</b>	<b>66</b>
<b>Lampiran 52. Akurasi Apriori Sentimen Latih (75:25).....</b>	<b>66</b>
<b>Lampiran 53. Akurasi Apriori Sentimen Latih (70:30).....</b>	<b>67</b>
<b>Lampiran 54. Akurasi Apriori Sentimen Latih (65:35).....</b>	<b>67</b>
<b>Lampiran 55. Akurasi Apriori Sentimen Uji (75:25).....</b>	<b>68</b>
<b>Lampiran 56. Akurasi Apriori Uji Sentimen (70:30).....</b>	<b>68</b>
<b>Lampiran 57. Akurasi Apriori Uji Sentimen (65:35).....</b>	<b>68</b>
<b>Lampiran 58. Hasil Penghitungan Akurasi Emotikon Sentimen Klasifikasi SVM (75:35) .....</b>	<b>68</b>
<b>Lampiran 59. Hasil Penghitungan Akurasi Emotikon Sentimen Klasifikasi SVM (70:30) .....</b>	<b>69</b>
<b>Lampiran 60. Hasil Penghitungan Akurasi Emotikon Sentimen Klasifikasi SVM (65:35) .....</b>	<b>69</b>



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Penggunaan gawai pintar atau *smartphone* sangat meningkat tajam pada masa kini. Sehingga berefek pada kondisi perekonomian masyarakat, khususnya pada proses memperoleh informasi finansial, yang berdampak pada naik turunnya suatu saham, mata uang negara maupun mata uang crypto. Klasifikasi sentimen pengguna atau *user* pada saat menjelajahi aplikasi *Twitter* yang berbasis online, dalam penelitian ini peneliti menggunakan metode *Apriori dan Support Vector Machines*. Metode *Apriori* ini menerapkan salah satu algoritma klasik data mining. Algoritma apriori digunakan agar komputer dapat mempelajari aturan asosiasi, mencari pola hubungan antar satu atau lebih item dalam suatu dataset. Metode Support Vector Machine (SVM) merupakan salah satu metode dalam supervised learning yang biasanya digunakan untuk klasifikasi (seperti Support Vector Classification) dan regresi (Support Vector Regression). Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan linier maupun non linear. Dewasa ini, meningkatnya penggunaan media sosial di masyarakat akan berdampak pada perkembangan opini publik serta meningkatnya peran berbagi informasi di media sosial. Oleh karena itu, dapat digunakan untuk menganalisis apakah opini publik itu positif, negatif, atau netral.

Metode menganalisis opini publik disebut analisis sentimen. Analisis sentimen adalah area penelitian yang menganalisis pendapat, opini, penilaian, penilaian, sikap, dan emosi individu tentang produk, layanan, organisasi, individu, masalah, peristiwa, topik, atribut, dan entitas lain (Bing Liu, 2012). Analisis sentimen adalah jenis pemrosesan bahasa alami, atau pemrosesan kata, untuk melacak emosi orang tentang produk atau opini tertentu. Saat melakukan analisis sentimen, komentar yang dianalisis dikategorikan ke dalam, dengan jelas menunjukkan kekuatan dan kelemahan produk gawai, memungkinkan perusahaan untuk meningkatkan kualitas produk dan kualitas layanan. Pada penelitian terdahulu (Hendry, et al., 2021, 26) yang mempunyai judul "*Analisis Sentimen pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machines (SVM)*" menyimpulkan Berdasarkan dari penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma Support Vector Machine (SVM) terbukti mampu diterapkan dengan baik untuk melakukan analisis sentimen dengan didukung oleh beberapa metode *preprocessing*, pembobotan term menggunakan TF-IDF, dan parameter *tuning* algoritma. Selain itu, pada proses pengujian algoritma Support Vector Machine (parameter tuning), nilai *accuracy* terbaik dihasilkan oleh kernel RBF dengan nilai parameter *complexity* = 10 dan gamma = 1, yaitu sebesar 84,37%. Sedangkan pada pengujian menggunakan 10-fold cross validation, diperoleh rata-rata nilai *accuracy* sebesar 80,41%. Nilai *precision*, *recall*, dan *f-score* terbaik juga dihasilkan oleh kernel RBF dengan nilai parameter *complexity* = 10 dan gamma = 1, yang masing-masing sebesar 84,33%, 84,67%, dan 84,50%. Penelitian berikutnya (Angelina, et al., 2020, 116) yang berjudul "*Analisis Sentimen*

*Aplikasi Ruang Guru di Twitter Menggunakan Algoritma Klasifikasi*” dapat disimpulkan analisis sentimen diambil dari komentar pengguna media sosial Twitter terhadap aplikasi Ruang Guru sebanyak 513 tweet, setelah dilakukan data cleaning, dengan sentimen positif sebanyak 338 tweet dan sentimen negatif sebanyak 175 tweet. Data tersebut diekstraksi menggunakan algoritma Naive Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbour (K-NN), dan *feature selection* dengan algoritma *Particle Swarm Optimization* (PSO). Penelitian ini membandingkan metode NB, SVM, K-NN tanpa menggunakan *feature selection* dengan metode NB, SVM, K-NN yang menggunakan *feature selection* serta membandingkan nilai *Area Under Curve* (AUC) dari metode-metode tersebut untuk mengetahui algoritma yang paling optimal. Hasil pengujian mendapatkan hasil bahwa aplikasi optimasi terbaik dalam model ini adalah algoritma PSO berbasis SVM dengan nilai akurasi sebesar 78,55% dan AUC sebesar 0,853. Penelitian ini berhasil mendapatkan algoritma yang efektif dan terbaik dalam mengklasifikasikan komentar positif dan komentar negatif terkait dengan aplikasi Ruang Guru. Penelitian selanjutnya (Tineges, et al, 2020., 650) yang berjudul “*Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM)*” dapat ditarik kesimpulan analisis sentimen tentang opini pengguna layanan Indihome pada Twitter, dengan tujuan untuk mendapatkan model klasifikasi sentimen menggunakan SVM, dan untuk mengetahui seberapa besar akurasi yang dihasilkan oleh metode SVM yang diterapkan pada analisis sentimen, serta untuk mengetahui seberapa puas pengguna layanan Indihome berdasarkan Twitter. Setelah dilakukan pengujian dengan metode SVM hasilnya adalah *accuracy* 87%, *precision* 86%, *recall* 95%, *error rate* 13%, dan *f1-score* 90%. Penelitian berikutnya (Aditya , 2020, 1) “*Algoritma Apriori Untuk Analisis Pola Pembelian Obat Konsumen*” dapat diambil kesimpulan proses berdasarkan pola pembelian obat dapat dilakukan dengan menerapkan metode algoritma apriori, pola pembelian obat berkunjung dapat dilakukan dengan melihat hasil kecenderungan konsumen untuk membeli obat kombinasi 3 *itemset*. Dengan menghitung *High Frequency Pattern Analysis* dan *Association Rule Formation*, *support* minimal 30% memiliki kombinasi 3 item yaitu MOLAGIT PER TAB (M1), VIT C TABLET (V2), dan PARACETAMOL 500 MG TABLET (P2) dengan *support* diperoleh hasil 33,33 %, dan *keyakinan* minimal 65% bahwa terdapat 6 aturan asosiasi final.

## 1.2 Rumusan Masalah

Pertukaran opini pengguna twitter terhadap informasi mengenai topik finansial sangat mempengaruhi seseorang. Hal ini dapat dilihat pada komentar yang yang dikirim di dalam aplikasi twitter, Pengaruh tersebut bisa berupa kondisi mental, dimana ketika mendengar kabar baik dari sisi finansial , maka seseorang mungkin akan terbawa arus untuk mengikuti trend finansial yang sedang terjadi. Sehingga penulis dapat membuat sebuah sistem informasi yang berbasis *Integrated Development Environment (IDE)* menggunakan bahasa *Python*, dan *IDE Google Co Lab* untuk menganalisa kondisi emosi seseorang dengan analisa sentimen.

### 1.3 Ruang Lingkup

Ruang Lingkup dari penelitian berikut adalah :

1. Data komentar pengguna yang menggunakan bahasa Inggris.
2. Data sentimen diambil dari komentar pada aplikasi sosial media Twitter dengan komentar yang berhubungan dengan finansial dengan topik perbincangan atau *keyword* yaitu mengenai bitcoin crypto, dan jenis klasifikasi sentimen yang diambil yaitu sentimen positif, hampir positif, negatif, hampir negatif dan netral.
3. Data Komentar twitter pada tahun 2021 bulan januari sampai dengan november 2022 dengan *Hashtags* : #crypto OR #bitcoin OR #cryptocurrency OR #blockchain OR #ethereum OR #btc OR #cryptonews OR #Tether OR #USDC OR #BNB

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Mengetahui tingkat emosi positif, negatif, dan netral terhadap komentar pengguna twitter termasuk dalam tentang topik mengenai finansial
2. Memilih komentar yang tidak sesuai dengan isi topik finansial dengan menjalankan proses *preprocessing*.
3. Menggunakan klasifikasi untuk mengukur akurasi analisis sentimen dari komentar pengguna aplikasi twitter yang beropini mengenai topik Finansial.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah membuat sistem yang dapat menganalisis sentimen komentar pengguna di aplikasi Twitter berupa kalimat dengan nilai positif, negatif, hampir positif, hampir negatif dan netral terkait topik Finansial. Untuk mengetahui kemampuan algoritma Apriori dan Support Vector Machines dalam menganalisis opini tentang topik finansial pengguna pada aplikasi twitter, ini adalah referensi lain bagi siapa saja yang tertarik dengan pengembangan.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### **2.1.1 Sistem Informasi**

Secara umum, sistem informasi adalah perangkat lunak yang dapat membantu mengatur atau menganalisis data. Tujuan utama dari sistem informasi adalah untuk mengubah data mentah menjadi informasi yang berguna bagi perusahaan atau instansi. (Raharja, 2022)

##### **2.1.2 Twitter**

Twitter adalah layanan *microblogging* online yang memungkinkan pengguna untuk mengirim, membaca, dan menanggapi pesan teks hingga 280 karakter (dikenal sebagai tweet). Pengguna tidak terdaftar di Twitter hanya dapat membaca tweet pengguna lain, sedangkan pengguna terdaftar dapat menulis, berbagi, dan menyukai tweet melalui antarmuka pengguna situs web dan aplikasi smartphone android dan iOS (iPhone). (Arifin, 2022)

##### **2.1.3 Algoritma Support Vector Machines (SVM)**

SVM adalah metode pembelajaran lanjutan yang menganalisis data dan memprediksi hasil. Algoritma ini dapat digunakan untuk klasifikasi dan analisis regresi. Namun, algoritma juga dapat membuat prediksi dan melakukan analisis pada sistem yang diberikan. Tujuan dari SVM adalah untuk memberikan informasi dari sejumlah besar kata dan untuk mengklasifikasikan data dengan label positif dan negatif. Manfaat SVM adalah dimensi input dan dimensi vektor. (Angelina Puput Giovani, 2020).

Algoritma SVM memiliki kelebihan dan kekurangan sebagai berikut:

Keuntungan:

- Kompatibilitas ruang dimensi tinggi
- Ketika jumlah dimensi melebihi jumlah sampel, metode ini efektif.
- Hemat memori karena menggunakan titik pelatihan fungsi keputusan (vektor pendukung)
- Ketika ada batas yang jelas antara kelas, itu bekerja dengan cukup baik.

Kelemahan

- Karena waktu pelatihan yang lama, algoritma SVM tidak cocok untuk kumpulan data yang besar.
- Ketika dataset mengandung lebih banyak *noise*, seperti kelas target yang tumpang tindih, SVM tidak bekerja dengan baik.
- SVM akan berkinerja buruk jika jumlah fitur untuk setiap titik data melebihi jumlah sampel data pelatihan.

- Tidak ada kejelasan probabilistik untuk klasifikasi karena pengklasifikasi vektor dukungan bekerja dengan menempatkan titik data di atas dan di bawah *hyperplane*. Hal ini dapat mengakibatkan beban komputasi yang signifikan.

#### 2.1.4 Algoritma Apriori

Metode apriori ini menerapkan salah satu algoritma klasik data mining. Algoritma apriori digunakan agar komputer dapat mempelajari aturan asosiasi, mencari pola hubungan antar satu atau lebih item dalam suatu *dataset*. (DQLAB, 2022)

Algoritma Apriori adalah sebuah metode bagi komputer untuk mempelajari aturan asosiasi dan menemukan pola hubungan antara satu atau lebih item dalam sebuah dataset. Algoritma ini sangat populer karena sering digunakan dalam penelitian data *mining*. (Dicky, 2020). Beberapa jenis algoritma yang tersedia saat ini, masing-masing dengan kelebihan dan kekurangannya sendiri. Tidak menjadi bingung dengan algoritma Apriori, algoritma ini memiliki banyak keuntungan dalam jangka panjang, tetapi memiliki beberapa kekurangan.

Kelebihan Algoritma Apriori :

1. Dibandingkan dengan algoritma lainnya, algoritma apriori dapat menangani data dalam jumlah besar.
2. Dapat menyederhanakan data.

Kekurangan Algoritma Apriori :

1. Memerlukan banyak waktu apabila memiliki banyak iterasi.
2. Dalam setiap iterasi memerlukan scan database.

#### 2.1.5 Analisis Sentimen

Analisis sentimen atau *opinion mining* adalah pengaplikasian dari ilmu komputasi linguistik, *text analysis*, dan *natural language processing* yang digunakan untuk mengekstrak subjektivitas dari sebuah sumber teks. Analisis sentimen sendiri kerap digunakan untuk menyatakan suka atau tidak suka terhadap suatu hal melalui sentimen positif dan negatif. Hal tersebut dapat dijadikan sebagai parameter untuk proses pengambilan keputusan. (LP2M, 2022)

#### 2.1.6 Text Mining

*Text Mining* adalah proses penemuan informasi dan pengetahuan dari teks. Dilakukan untuk menemukan pola untuk tujuan tertentu. Proses ini digunakan dalam analisis informasi dan pengambilan keputusan. Tahap awalnya adalah data *preprocessing* yang terdiri atas beberapa metode seperti *cleansing*, *case folding*, *tokenizing*, *filtering*, dan *stemming*. *Cleansing* adalah pembersihan data teks dari komponen tidak perlu, *case folding* adalah standarisasi teks menjadi *lower case*, *tokenizing* adalah pemecahan kalimat menjadi token, *filtering* adalah penghapusan *stopword*, dan *stemming* adalah transformasi kata-kata menjadi bentuk kata dasar.

#### 2.1.7 Python

Python adalah bahasa pemrograman yang digunakan untuk membuat aplikasi, mengelola komputer, dan melakukan analisis data. Python, sebagai bahasa tujuan

umum, dapat digunakan untuk membuat semua jenis program dan menyelesaikan masalah apa pun. (Algoritma, 2022).

### 2.1.8 Jupyter Notebook

*Jupyter Notebook* adalah aplikasi web sumber terbuka dan gratis yang memungkinkan Anda membuat dan berbagi dokumen dengan kode langsung, persamaan, visualisasi, dan teks naratif. Pembersihan dan transformasi data, simulasi numerik, pemodelan statistik, visualisasi data, dan pembelajaran mesin adalah semua aplikasi.

#### 2.1.8.1 NLTK

NLTK adalah perpustakaan pemrosesan data bahasa manusia.

#### 2.1.8.2 Numpy

Numpy adalah library yang berisi fungsi siap pakai untuk melakukan perhitungan ilmiah seperti matriks, aljabar, statistik, dan sebagainya.

#### 2.1.8.3 Korpus

Korpus adalah kumpulan besar teks, baik mentah atau dikelompokkan bersama.

## 2.2 Tinjauan Pustaka

Penelitian ini adalah penelitian yang menganalisis analisa sentimen. Tinjauan Pustaka yang disajikan berhubungan dengan penelitian berikut ini :

Penelitian tersebut yaitu penelitian yang dilakukan oleh Hendry Cipta Husada, Adi Suryaputra Paramita. Fakultas Teknik Universitas Suryaputra dengan judul “Analisis Sentimen Pada Maskapai Penerbangan di *platform* twitter menggunakan Algoritma Support Vector Machine (SVM)” .Berdasarkan dari penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma Support Vector Machine (SVM) terbukti mampu diterapkan dengan baik untuk melakukan analisis sentimen dengan didukung oleh beberapa metode *preprocessing*, pembobotan term menggunakan TF-IDF, dan parameter *tuning* algoritma. Selain itu, pada proses pengujian algoritma Support Vector Machine (parameter *tuning*), nilai akurasi terbaik dihasilkan oleh kernel RBF dengan nilai parameter *complexity* = 10 dan gamma = 1, yaitu sebesar 84,37%. Sedangkan pada pengujian menggunakan *10-fold cross validation*, diperoleh rata-rata nilai *accuracy* sebesar 80,41%. Nilai *precision*, *recall*, dan *f-score* terbaik juga dihasilkan oleh kernel RBF dengan nilai parameter *complexity* = 10 dan gamma = 1, yang masing-masing sebesar 84,33%, 84,67%, dan 84,50%.

## 2.3 Tabel Perbandingan Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Variabel Penelitian	Perbedaan
----	---------------	------------------	---------------------	-----------

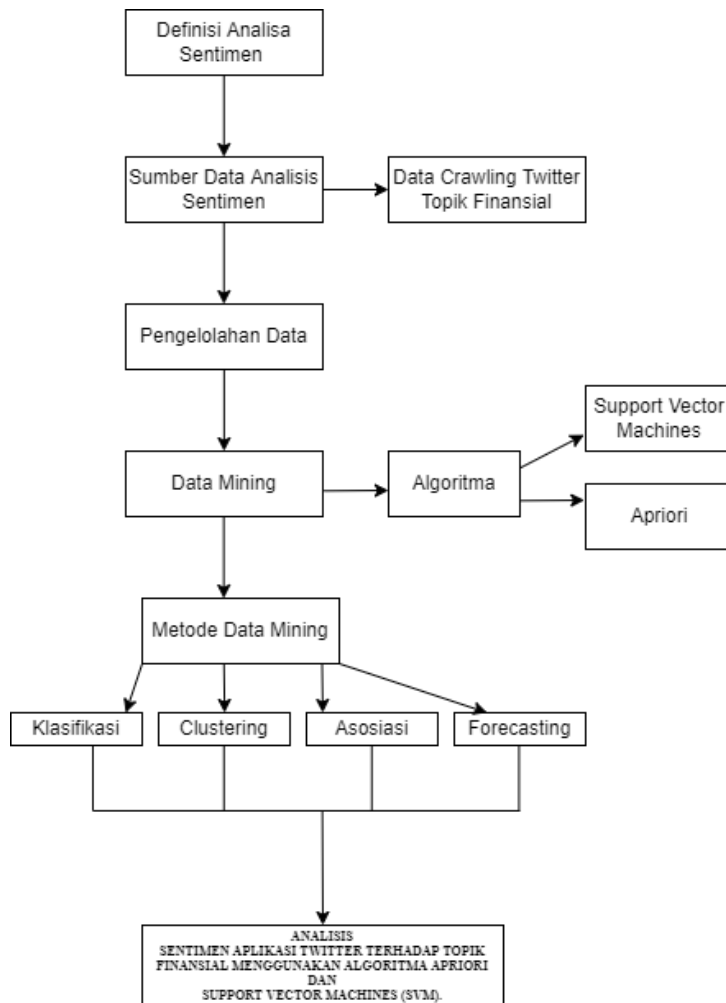


1.	Hendry Cipta Husada, Adi Suryaputra Paramita(2021)	Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)	-Karakteristik Komentar -Aplikasi Twitter -Topik Maskapai Penerbangan	-Periode Penelitian -Topik Finansial -Algoritma Apriori
2.	Angelina Puput Giovani, Ardiansyah, Tuti Haryanti, Laela Kurniawati, Windu Gata (2020)	Analisis Sentimen Aplikasi Ruang Guru di Twitter Menggunakan Algoritma Klasifikasi	-Karakteristik Komentar -Aplikasi Twitter -Topik Ruang Guru	-Periode Penelitian -Topik Finansial -Algoritma Apriori
3.	Rian Tineges, Agung Triayudi, Ira Diana Sholihati (2020)	Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM)	-Karakteristik Komentar -Aplikasi Twitter -Topik Indihome	-Periode Penelitian -Topik Finansial -Algoritma Apriori
4.	Aditya Dwi Prasetyo (2022)	Algoritma Apriori Untuk Analisis Pola Pembelian Obat Konsumen	-Karakteristik Komentar -Apotek Sekar Adi -Topik Pola Pembelian Obat Konsumen	-Topik Finansial -Aplikasi Twitter -Algoritma SVM
5	Ahmed Joko Susilo (2020)	Analisis Sentimen Ulasan Pengguna Aplikasi Kesehatan (Telemedicine) Sebagai Pendukung Keputusan Dalam Meningkatkan Kualitas Aplikasi Kesehatan	-Aplikasi Telemedicine -Topik Ulasan Pengguna Aplikasi	-Topik Finansial -Aplikasi Twitter -Algoritma SVM

## 2.4 Kerangka Teoritis

Kerangka teori penelitian diatas dapat di deskripsikan dan dapat dilihat pada gambar 1 bahwa pada proses data mining, dalam penelitian ini dibagi menjadi empat

yakni klasifikasi, *Clustering*, Asosiasi, dan *Forecasting*. Proses penelitian menggunakan metode data *mining* ini, menggunakan algoritma Apriori dan *Support Vector Machines* dan kemudian dapat dirumuskan kesimpulan mengenai judul dari penelitian ini, yang merupakan “Analisa Sentimen Aplikasi Twitter Terhadap Topik Finansial Menggunakan Algoritma Apriori dan *Support Vector Machines*.” Penelitian berikut tidak hanya menjelaskan bagaimana suatu data diolah untuk digunakan dalam analisis sentimen saja.



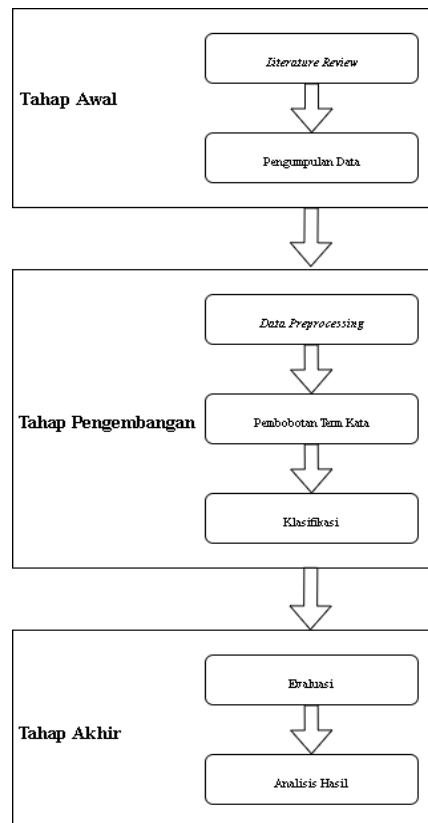
**Gambar 1. Flowchart Kerangka Teoritis**

Melainkan juga menjelaskan mengenai definisi dari data mining dan analisa sentimen, serta implementasi dari algoritma terkait yang digunakan dalam penelitian ini. Deskripsi atau penjelasan teori yang peneliti tulis memiliki tujuan untuk para pembaca laporan penelitian ini, yaitu para pembaca mengetahui bagaimana cara mengolah data untuk melakukan analisis sentimen sesuai dengan ketentuan dari kaidah data *mining* dan algoritma dalam penelitian ini. Sehingga di masa mendatang bisa menyiapkan solusi analisis sentimen yang lebih baik dari segi penelitian yang penulis tulis dalam laporan penelitian ini.

## BAB III METODE PENELITIAN

### 3.1 Metode Penelitian

Penelitian ini menggunakan metode, setidaknya dipecah menjadi 3 tahap utama, yaitu tahap awal, tahap pengembangan, dan tahap akhir. Setiap tahapan proses terdiri dari beberapa langkah. Seluruh proses dapat dilihat pada gambar 2 di bawah ini.



**Gambar 2. Diagram Alir Metode Penelitian** (Hendry Cipta Husada, 2021)

### 3.2 Literatur Review

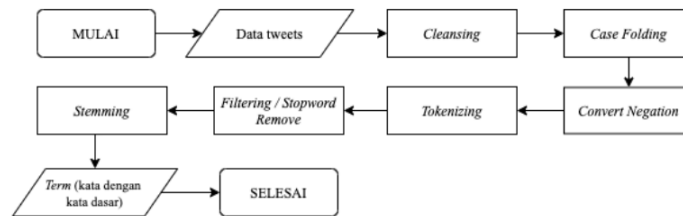
Pencarian sumber-sumber tulisan sehubungan dengan penyelidikan asumsi yang dapat mendukung penyelidikan ini secara hipotesis dalam proses analisis ini, dilakukan. Tulisan yang didapat berasal dari buku, skripsi, jurnal, dan sumber perpustakaan dari web

### 3.3 Pengumpulan Data

Dataset yang digunakan pada penelitian ini adalah kumpulan opini berupa tweet terhadap beberapa akun official yang membahas *tweet* finansial di twitter di beberapa negara yang menggunakan bahasa Inggris pada periode 2022 yang diperoleh menggunakan website kaggle. Semua data tweet disimpan dalam format *.csv*.

### 3.4 Data Preprocessing

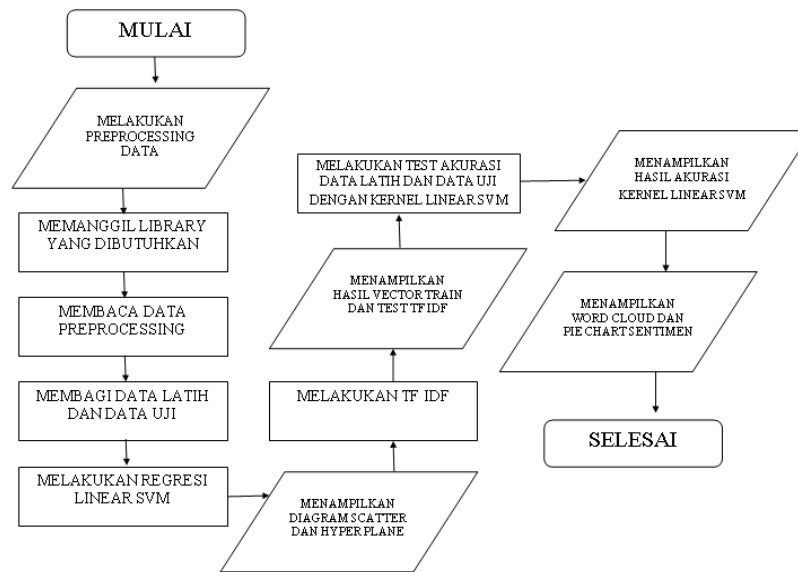
Kumpulan data tweet harus melalui tahapan *preprocessing* untuk melakukan transformasi data yang awalnya berbentuk tidak terstruktur atau terdapat banyak *noise* sebelum masuk dalam proses klasifikasi. Menjadi data yang terstruktur sehingga proses analisis menjadi lebih mudah. Beberapa metode yang akan diterapkan dalam proses data *preprocessing* dapat dilihat pada gambar 3.



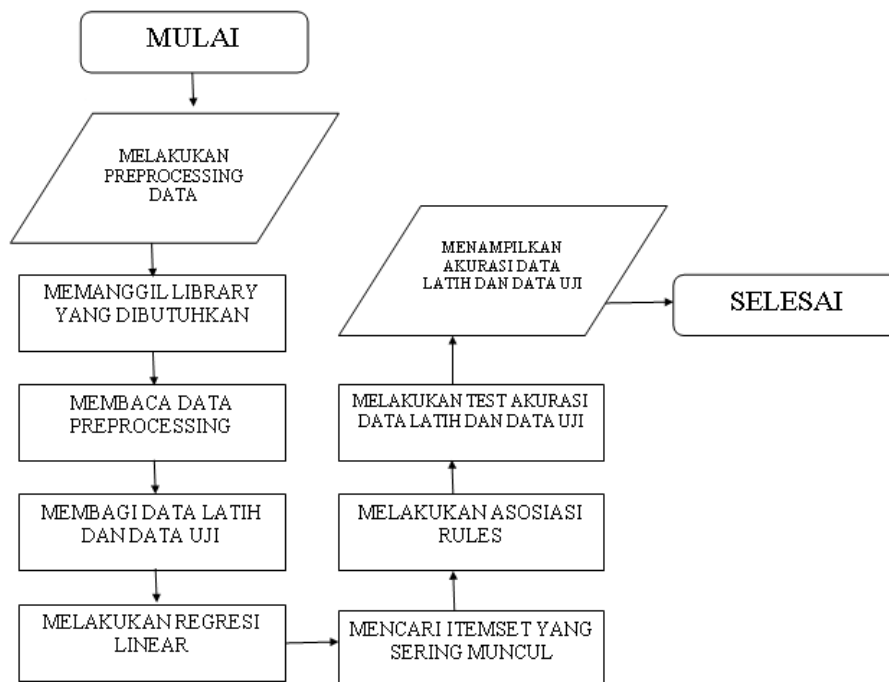
Gambar 3. Diagram Alir Data *Preprocessing* (Hendry Cipta Husada, 2021)

### 3.5 Pembobotan Term/Kata

Proses ini, kumpulan term/kata akan di vektorisasi atau diubah menjadi bentuk numerik. Hal tersebut perlu untuk dilakukan karena dalam klasifikasi hanya dapat memproses data dalam bentuk numerik saja. Proses pembobotan dalam penelitian ini menggunakan metode *Term Frequency-Inverse Document Frequency* (TF- IDF). Alur proses pembobotan dengan TF-IDF dan *association rule* dapat dilihat pada gambar 4 dan gambar 5.



Gambar 4. Flowchart Algoritma SVM (Husada, 2021)



Gambar 5. Flowchart Algoritma Association Rule (Husada, 2021)

### 3.6 Klasifikasi Data

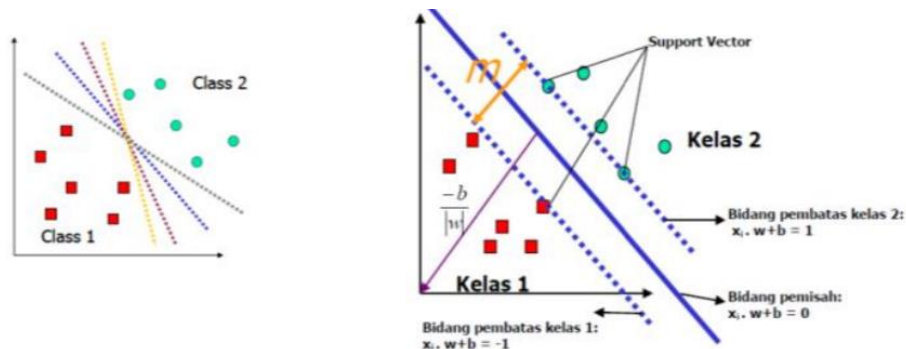
Klasifikasi data dalam penelitian ini dilalui melalui dua tahapan yaitu:

- Tahapan pertama menggunakan klasifikasi *Support Vector Machine* (SVM)
- Tahapan kedua menggunakan klasifikasi *Association Rule* atau algoritma Apriori



### 3.6.1 Klasifikasi Support Vector Machine (SVM)

Klasifikasi algoritma Support Vector Machine (SVM) digunakan untuk mengenali pola dalam data pelatihan berlabel, dan kemudian menghasilkan model pembelajaran mesin. Model yang dihasilkan kemudian digunakan untuk mengidentifikasi data tweet baru yang tidak berlabel untuk menghasilkan prediksi kelas emosi, yaitu positif, negatif, hampir positif, hampir negatif dan netral. Contoh Implementasi perhitungannya adalah sebagai berikut :



**Gambar 6. Hyperplane Support Vector Machine** (Husada, 2021)

Secara matematika konsep SVM dapat dijelaskan sebagai berikut, data yang tersedia dinotasikan  $x \in R^d$  sedangkan label setiap kelas dinotasikan  $y_i \in \{-1, +1\}$  untuk  $i = 1, 2, 3, \dots, n$ . Diasumsikan kedua kelas objek dapat terpisah secara sempurna oleh *hyperplane* berdimensi  $d$ , yang didefinisikan (Praptiwi, 2018):

$$w^{\rightarrow} \cdot x + b = 0 \quad (2.4)$$

Pola  $x$ , yang termasuk dalam kelas  $+1$  (sampel positif) dapat didefinisikan sebagai berikut :

$$w^{\rightarrow} \cdot x + b \geq +1 \quad (2.5)$$

Pada pola  $x$ , yang termasuk kelas  $-1$  (sampel negatif) dinotasikan sebagai berikut :

$$w^{\rightarrow} \cdot x + b \leq -1 \quad (2.6)$$

Dengan keterangan sebagai berikut :

$R^d$  = ruang vektor

$w^{\rightarrow}$  = vektor bobot

$x$  = vektor data (input)

$b$  = bias

Pada pemisahan *hyperplane* dengan bentuk canonical harus memenuhi *constraint* atau batasan yang disebut sebagai fungsi kendala, dengan notasi sebagai berikut :

$$y_i [(w^{\rightarrow} \cdot x) + b] \geq 1, \quad i = 1, \dots, l \quad (2.7)$$

Kemudian *hyperplane* yang memisahkan data harus meminimalkan,

$$\min w = \tau(w) = \frac{1}{2} \|w\|^2 \quad (2.8)$$

Support Vector Machine (SVM) biasanya ditemui kondisi dimana kernel linear bekerja tidak optimal yang mengakibatkan hasil klasifikasi terhadap data menjadi buruk dalam proses klasifikasi dengan. Hal tersebut dapat diatasi dengan menggunakan kernel non-linear dengan memanfaatkan kernel trick. Dengan memanfaatkan kernel trick, akan dilakukan mapping data input ke *feature space* yang dimensinya lebih tinggi sehingga membuat data input yang dihasilkan akan terpisah secara linear dan membentuk *hyperplane* yang optimal . Persamaan dari setiap kernel SVM dapat dilihat pada Tabel 1 .

**Tabel 1. Persamaan Setiap Kernel SVM (Husada, 2021)**

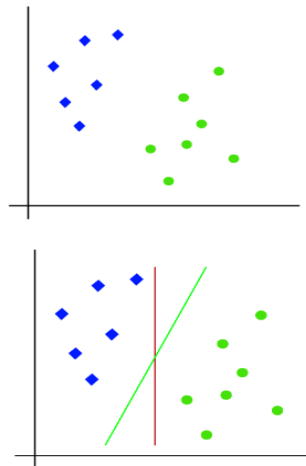
Kernel	Persamaan
<i>Linear</i>	$K(x_i, x) = x_i^T x$
<i>Polynomial</i>	$K(x_i, x) = (\gamma \cdot x_i^T x + r)^p, \gamma > 0$
<i>RBF</i>	$K(x_i, x) = \exp \exp (-\gamma  x_i - x ^2), \gamma > 0$
<i>Sigmoid</i>	$K(x_i, x) = \tanh (\gamma x_i^T x + r)$

### 3.6.1.1 Jenis-jenis Algoritma SVM

SVM diklasifikasikan menjadi dua jenis:

#### 1. SVM Linear

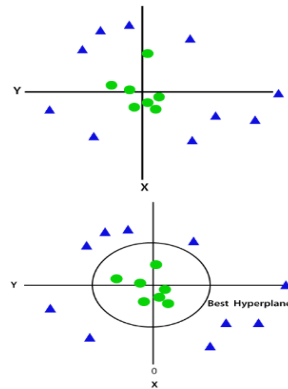
Linear SVM digunakan untuk data yang dapat dipisahkan secara linier, artinya jika suatu *dataset* dapat diklasifikasikan menjadi dua kelas dengan menggunakan satu garis lurus, maka data tersebut dikenali sebagai data yang dapat dipisahkan secara linier, dan *classifier* yang digunakan dikenal sebagai *classifier* SVM Linear. Contoh tampilan data linier dengan pada gambar 7.



**Gambar 7. Grafik SVM Linear (Trivusi, 2022)**

#### 2. Non-linear SVM

Non-linear SVM digunakan untuk data yang dapat dipisahkan secara non-linear, artinya jika suatu dataset tidak dapat diklasifikasikan menggunakan garis lurus maka dianggap data non-linear, dan *classifier* yang digunakan adalah non-linear SVM *classifier*. Contoh tampilan data non-linear dengan pada gambar 8.



**Gambar 8. SVM Non-linear** (Trivusi, 2022)

### 3.6.2.2 TF-IDF

*Term Frequency — Inverse Document Frequency* atau TF — IDF adalah suatu metode algoritma yang berguna untuk menghitung bobot setiap kata yang umum digunakan. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat. Metode ini akan menghitung nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap token (kata) di setiap dokumen dalam korpus. Secara sederhana, metode TF-IDF digunakan untuk mengetahui seberapa sering suatu kata muncul di dalam dokumen. (Sierra, 2019)

Untuk menentukan frekuensi kemunculan kata dalam dokumen. Akibatnya, semakin besar frekuensi kemunculan kata tersebut, semakin besar nilainya. Ada beberapa jenis rumus yang dapat digunakan dalam *Term Frequency* (TF):

1. TF *binary* (TF biner) hanya mempertimbangkan ada atau tidaknya suatu kata atau term dalam dokumen; jika ya, itu diberi nilai satu (1); jika tidak, itu diberi nilai nol (0).
2. TF Murni (raw TF), nilai TF ditentukan oleh berapa kali suatu term muncul dalam dokumen. Misalnya, jika muncul lima (5) kali, kata tersebut bernilai lima poin (5).
3. Normalisasi TF, yang melibatkan perbandingan frekuensi istilah dengan nilai maksimum dari keseluruhan atau kumpulan istilah frekuensi dalam dokumen.
4. Logarithmic TF, untuk menghindari dominasi dokumen dengan sedikit term dalam query tetapi dengan frekuensi yang tinggi.

### 3.6.2 Klasifikasi Apriori

*Association rule* merupakan salah satu dari tugas atau teknik yang terdapat dalam data *mining*. Tugas *association rule* adalah untuk menemukan atribut yang muncul secara bersamaan atau aturan asosiatif antara suatu kombinasi item. Pada dunia bisnis, teknik ini lebih dikenal dengan analisis afinitas atau analisis keranjang belanja (market basket analysis) *association rule* adalah bentuk jika “kejadian sebelumnya” kemudian

“konsekuensinya” (IF antecedent, THEN consequent) dengan ukuran *support* dan *confidence* terkait dengan aturan tersebut (Larose & Larose, 2014). *Support* dan *confidence* merupakan bagian dalam pengukuran aturan ketertarikan (rule interestingness), karena kedua ukuran tersebut mencerminkan kegunaan dan kepastian aturan yang ditemukan (Han, Kamber, & Pei, 2012). Hal tersebut menjadikan parameter yang digunakan dalam *association rule* (Sheikh, Tanveer, & Hamdani, 2004). Berikut merupakan contoh perhitungan algoritma apriori untuk analisis sentimen :

**Tabel 2. Tabel Kata Asosiasi Rule**

No	Kata
1	'price', 'update', 'usd', 'follow', 'for', 'recent', 'eth', 'price', 'updates'
2	'do', 'well', 'to', 'understand', 'that', 'every', 'dip', 'in', 'the', 'price', 'of', 'provides', 'golden', 'opportunity', 'to', 'accumulate', 'ones', 'you', 'understand', 'this', 'youll', 'know', 'to', 'do', 'buyhodlbe', 'patient', 'follow', 'shes', 'always', 'available', 'during', 'ups', 'and', 'downher', 'tweetstapredictions', 'amp', 'tips', 'are', 'so', 'accurate'
3	'kayla', 'harold', 'egbert', 'laverna'
4	'yesterday', 'we', 'made', 'slam', 'buyback', 'and', 'locked', 'it', 'forever', 'in', 'addition', 'to', 'the', 'slam', 'weve', 'locked', 'in', 'the', 'last', 'months', 'see', 'you', 'at', 'another', 'burning', 'fest'

Dengan memperhitungkan *itemset* {price}, kita bisa menentukan support dari asosiasi *rule* {price} -> {you}. *support* dapat ditemukan dengan membagi jumlah transaksi yang memuat *itemset* {price} -> {you} dengan jumlah total transaksi.

Dalam hal ini, nilai support dari asosiasi *rule* {price} dan {you} adalah.

$$= \text{Support } (X \rightarrow Y)$$

$$= \text{Jumlah transaksi yang mengandung item X dan Y} / \text{Jumlah total transaksi}$$

$$= 2/4$$

$$= 0,5$$

- Dengan demikian, nilai *confidence* dari asosiasi *rule* {price} dan {you} adalah:

$$\text{Confidence} = (\text{Jumlah transaksi yang memuat item } \textit{antecedent} \text{ dan } \textit{consequent}) / (\text{Jumlah transaksi yang memuat item } \textit{antecedent})$$

$$= 0 / 2$$

$$= 0 \text{ (karena dalam empat transaksi jumlah } \textit{consequent} \text{ dari kata “price” tidak ada)}$$

### 3.6.2.1 Support

*Support* ditentukan oleh jumlah item dalam koleksi dan memberikan persentase transaksi yang mengandung  $Z$ , dan digunakan sebagai ukuran signifikansi (kepentingan) untuk koleksi (Agrawal, Imielinski, & Swami, 1993). Ini adalah representasi dari kombinasi item tertentu yang muncul dalam serangkaian transaksi atau dalam *database*.

Perhitungan dalam menentukan nilai support dapat didefinisikan seperti pada rumus dibawah ini.

$$\text{Support} = P(A \cap B) = \frac{\text{Jumlah transaksi memuat } A \text{ dan } B}{\text{Total jumlah transaksi}}$$

### 3.6.2.2 Confidence

*Confidence* didefinisikan sebagai kemungkinan mengamati peristiwa berikutnya dengan pengetahuan bahwa transaksi juga mengandung anteseden (Agrawal, Imielinski, & Swami, 1993). Nilai *confidence* ini merupakan hubungan yang kuat antar item dalam aturan asosiasi (association rule) yang terbentuk. Perhitungan dalam nilai penentu keyakinan dapat didefinisikan dengan

$$\text{Confidence} = P(B|A) = \text{Support}(A \cap B) / P(A) \quad (3.2)$$

## 3.7 Evaluasi

Proses selanjutnya setelah klasifikasi adalah melakukan evaluasi untuk mengecek performansi model algoritma *Support Vector Machine* (SVM) dan Apriori. Evaluasi dilakukan dengan menggunakan alat yang disebut matriks konfusi untuk mengukur nilai *accuracy*, *precision*, *recall*, dan *f-score* pada SVM dan nilai *accuracy* pada Apriori.

## 3.8 Analisis Hasil

Analisis deskriptif terhadap hasil pengujian algoritma *Support Vector Machine* (SVM) dan algoritma Apriori untuk melakukan klasifikasi sentimen dengan membandingkan hasil akurasi yang paling tinggi dan rendah. Analisis deskriptif adalah suatu teknik yang digunakan untuk mengevaluasi dan mempresentasikan data secara sederhana dan jelas. Dalam hal ini, analisis deskriptif dilakukan terhadap hasil pengujian algoritma *Support Vector Machine* (SVM) dan algoritma Apriori untuk melakukan klasifikasi sentimen. Tujuannya adalah untuk membandingkan hasil akurasi dari kedua algoritma dan menentukan algoritma mana yang memiliki hasil akurasi paling tinggi dan rendah. Algoritma SVM dan Apriori akan diuji dengan data sentimen untuk menentukan hasil akurasi dari masing-masing algoritma. Kemudian, hasil akurasi akan dibandingkan untuk menentukan algoritma mana yang memiliki hasil akurasi paling tinggi dan rendah.

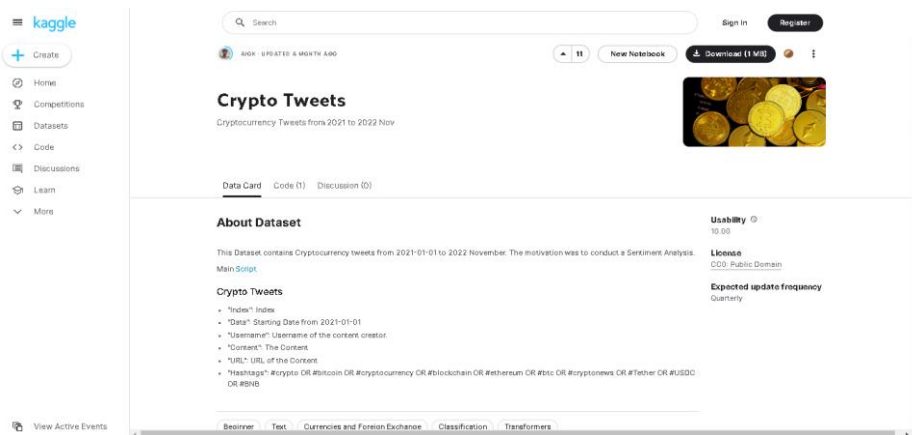


## BAB IV RANCANGAN DAN IMPLEMENTASI

### 4.1 Tahap Awal

#### 4.1.1. Proses Pencarian Data

Data yang dipakai dalam penelitian ini adalah data komentar twitter pada akun perusahaan dan *influencer* yang membicarakan mengenai finansial. Data tersebut diperoleh melalui website bernama “Kaggle” yang menyediakan *dataset* komentar twitter pada tahun 2021 – 2022 November dengan *Hashtags* : : #crypto OR #bitcoin OR #cryptocurrency OR #blockchain OR #ethereum OR #btc OR #cryptonews OR #Tether OR #USDC OR #BNB. Gambar 9 dibawah ini Memperlihatkan tampilan dari website “Kaggle”.



**Gambar 9. Tampilan Dashboard Website Kaggle (leoth9, 2022)**

Pencarian dan pengumpulan data pada awalnya dengan menuliskan *keyword* pada bagian *search*. Setelah dilakukan tahapan tadi maka akan muncul kumpulan data dari berbagai macam sumber yang merupakan hasil *scraping* oleh para data analis dalam *website* tersebut. Data yang sudah diperoleh dapat langsung di *download*, dengan memilih pada bagian simbol tanda panah kebawah dalam format *.csv*. Jumlah total komentar yang diperoleh dari beberapa akun tersebut adalah 10000 komentar. Data dapat dilihat dalam gambar 10 dibawah ini.

Out [6]:

	no	Date	id	text	URL	Hashtags
0	0	2022-11-30 11:53:21+00:00	0xEthereumYoda	#Ethereum price update : \$1.269.23 USD...	https://twitter.com/0xEthereumYoda/status/1597...	[Ethereum, 'ETH', 'Bitcoin', 'BTC', 'altcoin...
1	1	2022-11-30 11:53:21+00:00	Lawrenc32984128	@mtmalinen @ecb Do well to understand that eve...	https://twitter.com/Lawrenc32984128/status/159...	[BTC]
2	2	2022-11-30 11:53:21+00:00	NITESH5784410	Kayla #Ethereum Harold #世界杯 Egbert #百家乐 Lavern...	https://twitter.com/NITESH5784410/status/159...	[Ethereum, '世界杯', '百家乐', '世界杯竞猜]
3	3	2022-11-30 11:53:20+00:00	817coin	#Bitcoin https://co2koLcVcN https://	https://twitter.com/817coin/status/15979217174...	[Bitcoin]
4	4	2022-11-30 11:53:19+00:00	slamtoken	Yesterday we made a 3,000 SLAM buyback and lo...	https://twitter.com/slamtoken/status/159792171...	[slamtoken, 'bnb', 'bsc]
...	...	...	...	...	...	...
9995	9995	2022-11-30 09:16:43+00:00	tecniaofficial	Digital Rupee or e-Rupee is a form of digital ...	https://twitter.com/tecniaofficial/status/1597...	[Rias, 'tecnia', 'rupee', 'digital', 'digita...
9996	9996	2022-11-30 09:16:41+00:00	BullshitTrader_	In bear market, short every pump n dca every d...	https://twitter.com/BullshitTrader_/status/159...	[btc, 'bitcoin', 'crypto]
9997	9997	2022-11-30 09:16:40+00:00	rekIBart	Take that high and I'm max short #btc https://	https://twitter.com/rekIBart/status/1597882292...	[btc]
9998	9998	2022-11-30 09:16:39+00:00	kamalaishehabi	في هذه الفوج حول العالم، يتعمد على م...	https://twitter.com/kamalaishehabi/status/1597...	[crypto, 'binance', 'تعمد]
9999	9999	2022-11-30 09:16:38+00:00	jimmymys24	The 15m #bitcoin just went into self custody h...	https://twitter.com/jimmymys24/status/159788228...	[bitcoin]

10000 rows x 6 columns

Gambar 10. Dataset Financial Tweet (leoth9, 2022)

## 4.2 Tahap Pengembangan

### 4.2.1. Analisis Deskriptif

Data yang didapatkan, dilakukan proses *cleaning*/pembersihan. Tujuan dari prosedur ini adalah untuk menghapus data yang sebelumnya diperoleh melalui *scraping* oleh para data analis. Data yang disimpan dan tetap digunakan adalah kolom *id*, *text*, *sentimen*, *polar\_sentimen*, *subjectivity*, *sentimen\_emot*, *text\_tokens*, *stem\_text*, *text\_tokens\_WSW*, dan *text\_tokens\_fdist* di lampiran. Setelah dilakukan *cleaning* pada data, kemudian data divisualisasikan untuk memudahkan dalam proses analisis deskriptif. Diagram batang menjadi opsi pilihan untuk memvisualisasikan data tersebut. Visualisasi ini dilakukan untuk melihat informasi yang di dapat dari data sentimen twitter. Aspek total jumlah dari komentar sentimen *twitter* diperoleh melalui visualisasi data berdasarkan *id* dan *text*. Pada gambar 11 dibawah ini merupakan *pie chart* mengenai komentar pada *sentiment* crypto bitcoin, untuk klasifikasi sentimen *text* dan sentimen emotikon yang di tampilkan yaitu positif, negatif, netral, hampir positif, dan hampir negatif, tampilan data klasifikasi sentimen tersebut dari sentimen tersebut.

	id	text	sentimen	polar_sentimen	subjectivity	sentimen_emot	text_tokens	stem_text	text_tokens_W	text_tokens_fdist
1	0	0xEthereu... etherium	netral	0	0.25	Netral	[Ethereum, 'Ethereum', 'Ethereum', ...]			
2	1	Lawrenc32984128	positif	0.36666667	0.51111111	Positif	[mtmalinen, 'mtmalinen', 'mtmalinen', ...]			
3	2	NITESH5784410	netral	0	0	Netral	[kayla, 'kayla', 'kayla', 'kayla', ...]			
4	3	817coin	netral	0	0	Netral	[bitcoin, 'bitcoin', 'bitcoin', ...]			
5	4	slamtoken	netral	0	0.06666667	Negatif	[yesterday, 'yesterday', 'yesterday', ...]			
6	5	JenniferVakay	positif	0	0	Positif	[kay, 'kay', 'kay', 'kay', ...]			
7	6	futuristic_	hampir negatif	-0.15	0.76666667	Hampir Negatif	[as, 'som', 'as', 'som', 'someone', ...]			
8	7	Amit76321	netral	0	0	Netral	[stewart, 'stewart', 'stewart', ...]			
9	8	Darabehn	positif	0.4	0.3	Hampir Positif	[, 'ma', 'ma', 'ma', 'ma', ...]			
10	9	blodochan	hampir positif	0.10771609	0.37375402	Positif	[the, 'the', 'the', 'the', 'the', ...]			
11	10	itsalansh	hampir positif	-0.14333333	0.77333333	Positif	[themoor, 'themoor', 'themoor', ...]			
12	11	NFTSiversi	netral	0	0	Netral	[crypto, 'crypto', 'crypto', ...]			
13	12	WhaleCoi	netral	0	0	Netral	[bitcoin, 'bitcoin', 'bitcoin', ...]			
14	13	Sakshy73	netral	0	0	Netral	[darlene, 'darlene', 'darlene', ...]			
15	14	Ashishk26	hampir positif	0.13636364	0.45454545	Netral	[virtual, 'virtual', 'virtual', ...]			
16	15	nextintell	netral	0	0	Netral	[idelle, 'idelle', 'idelle', 'idelle', ...]			
17	16	crypto_who	positif	0.5	0.5	Hampir Positif	[top, 'top', 'top', 'top', 'top', ...]			
18	17	ManishJ5	netral	0	0	Netral	[hobart, 'hobart', 'hobart', ...]			
19	18	Pratin93	netral	0	0	Netral	[todd, 'todd', 'todd', 'todd', ...]			


Gambar 11. Tampilan Data Klasifikasi Sentimen

### 4.2.2. Sentimen

Data yang tersedia di situs web Kaggle bersifat tekstual. Data tersebut masih memiliki kekurangan, seperti tidak terstruktur dan mengandung angka, tanda baca, emotikon, dan kata-kata yang tidak bermakna. Akibatnya, *preprocessing* dokumen teks sebelum klasifikasi menurut kelas sentimen harus dilakukan secara bertahap. Tahapan ini berupaya untuk membakukan bentuk kata atau teks pada setiap ulasan pengguna dan menghilangkan karakter selain huruf. Ada beberapa langkah yang dapat dilakukan selama tahap *preprocessing*. Langkah-langkah yang dilakukan dalam penelitian ini adalah pembersihan data dan pelipatan kasus, yang berarti normalisasi, *stemming*, *tokenizing*, dan *stop word removal*.

### 4.2.3. Data Cleaning dan Case Folding

Tahap pembersihan data bertujuan untuk menghapus data yang tidak terdapat teks pada ulasan dan membersihkan ulasan/komentar dari item yang tidak diperlukan. Dokumen teks dengan tanda baca, angka, dan emotikon harus dibersihkan. Selama tahap pelipatan huruf, setiap huruf dalam dokumen teks diubah menjadi huruf kecil. Pada gambar 12 dibawah menampilkan proses pembersihan data.



	id	text
0	0xEthereumYoda	#Ethereum price update: '\n\n#ETH \$1269.23 USD\...
1	Lawrenc32984128	@mtmalinen @ecb Do well to understand that eve...
2	NITESHP55784410	Kayla #Ethereum Harold #世界杯 Egbert #百家乐 Lavern...
3	817coin	#Bitcoin https://t.co/2koLICvCri https://t.co/...
4	slamtoken	Yesterday we made a 3,000 SLAM buyback and loc...

	id	text
0	0xEthereumYoda	price update usd follow for recent eth price u...
1	Lawrenc32984128	do well to understand that every dip in the pr...
2	NITESHP55784410	kayla harold egbert laverna
3	817coin	
4	slamtoken	yesterday we made slam buyback and locked it ...

**Gambar 12. Proses *Cleaning* Data**

Setelah proses menghilangkan data, jumlah data yang akan dilanjutkan pada proses berikutnya menjadi tereduksi. Jumlah data yang akan digunakan adalah 10.000 ulasan atau komentar. Ulasan / komentar yang masih memiliki angka, tanda baca, dan emoticon, kemudian dibersihkan. Proses membersihkan dibantu dengan *library* yang

terdapat dalam bahasa pemrograman python. Pada *library* tersebut juga disediakan *library* untuk mengubah huruf besar menjadi huruf kecil. Proses tersebut dapat diketahui pada Tabel 3 .

**Tabel 3. Proses *cleaning data***

No	Sebelum	Sesudah
1	#Ethereum price update: #ETH \$1269.23 USD #Bitcoin 0.07510228 #BTC Follow for recent ETH price updates #altcoin, #cryptocurrency, #crypto	price update usd follow for recent eth price up- dates
2	@mtmalinen @ecb Do well to understand that every dip in the price of #BTC provides a golden opportunity to accumulate .Ones you un- derstand this, you'll know to do: Buy,HODL,Be patient.follow @btcskyy,she's always available during ups and down,her tweets,TA,predictions & tips are so accurate	do well to understand that every dip in the price of provides golden oppor- tunity to accumulate ones you understand this youll know to do buyhodlbe pa- tient follow shes always available during ups and downher tweetsta predic- tions amp tips are so accu- rate
3	Kayla #Ethereum Harold #世界杯 Egbert #百家乐 Laverna #世界杯买球 <a href="https://t.co/smw7VJ19WK">https://t.co/smw7VJ19WK</a>	kayla harold egbert la- verna
4	#Bitcoin <a href="https://t.co/2koLICvCri">https://t.co/2koLICvCri</a> <a href="https://t.co/MTR3woxLb9">https://t.co/MTR3woxLb9</a>	-

#### 4.2.4. Tokenizing

*Tokenizing* adalah proses pemisahan kata-kata dalam dokumen teks menjadi kata-kata yang berdiri sendiri atau tidak berhubungan (Praptiwi, 2018). Token adalah penggalan kata atau kata yang telah dipisahkan. Setiap kata dalam pemisahan kata ini akan dipisahkan dengan tanda koma (,).Tabel 4 menunjukkan contoh *tokenizing*.

**Tabel 4. Proses tokenisasi data.**

No	Sebelum	Sesudah
1	price update usd follow for recent eth price up- dates	'price', 'update', 'usd', 'fol- low', 'for', 'recent', 'eth', 'price', 'updates'
2	do well to understand that every dip in the price of provides golden opportunity to accumulate	'do', 'well', 'to', 'under- stand', 'that', 'every', 'dip',

	ones you understand this youll know to do buyhodlbe patient follow shes always available during ups and downher tweetstapredictions amp tips are so accurate	'in', 'the', 'price', 'of', 'provides', 'golden', 'opportunity', 'to', 'accumulate', 'ones', 'you', 'understand', 'this', 'youll', 'know', 'to', 'do', 'buyhodlbe', 'patient', 'follow', 'shes', 'always', 'available', 'during', 'ups', 'and', 'downher', 'tweetstapredictions', 'amp', 'tips', 'are', 'so', 'accurate'
3	kayla harold egbert laverna	'kayla', 'harold', 'egbert', 'laverna'
4	yesterday we made slam buyback and locked it forever in addition to the slam weve locked in the last months see you at another burning fest	'yesterday', 'we', 'made', 'slam', 'buyback', 'and', 'locked', 'it', 'forever', 'in', 'addition', 'to', 'the', 'slam', 'weve', 'locked', 'in', 'the', 'last', 'months', 'see', 'you', 'at', 'another', 'burning', 'fest'

#### 4.2.5. Stop Word Removal

Tahap *stop word removal* menghilangkan kata-kata yang tidak perlu tanpa mempengaruhi sentimen. Pustaka bahasa Inggris, yang tersedia dalam bahasa pemrograman Python, digunakan untuk menghapus kata-kata ini. Perpustakaan ini memiliki koleksi kata-kata stopword dalam bahasa Inggris yang mematuhi aturan *stop-words* Python. Setelah data diberi token, data tersebut dapat mengalami proses penghapusan kata berhenti. Hal ini disebabkan karena data yang digunakan merupakan kumpulan besar dokumen teks (komentar per baris). Karena sudah berbentuk token, akan memudahkan perpustakaan bahasa Inggris untuk mengecek setiap kata. Contoh *stop-words removal* dapat dilihat pada tabel 5.

**Tabel 5. Proses Stop Word Removal**

No	Sebelum	Sesudah
1	'price', 'update', 'usd', 'follow', 'for', 'recent', 'eth', 'price', 'updates'	'price', 'updat', 'usd', 'follow', 'recent', 'eth', 'price', 'updat'
2	'do', 'well', 'to', 'understand', 'that', 'every', 'dip', 'in', 'the', 'price', 'of', 'provides', 'golden', 'opportunity', 'to', 'accumulate', 'ones', 'you', 'understand', 'this', 'youll', 'know', 'to', 'do', 'buyhodlbe',	'well', 'understand', 'everi', 'dip', 'price', 'provid', 'golden', 'opportun', 'accumul', 'one', 'understand',



	'patient', 'follow', 'shes', 'always', 'available', 'during', 'ups', 'and', 'downher', 'tweetstapredictions', 'amp', 'tips', 'are', 'so', 'accurate'	'thi', 'youll', 'know', 'buyhodlb', 'patient', 'follow', 'alway', 'avail', 'dure', 'downher', 'tweetstapredict', 'amp', 'tip', 'accur'
3	'kayla', 'harold', 'egbert', 'laverna'	'kayla', 'harold', 'egbert', 'laverna'
4	'yesterday', 'we', 'made', 'slam', 'buyback', 'and', 'locked', 'it', 'forever', 'in', 'addition', 'to', 'the', 'slam', 'weve', 'locked', 'in', 'the', 'last', 'months', 'see', 'you', 'at', 'another', 'burning', 'fest'	'yesterday', 'made', 'slam', 'buyback', 'lock', 'forev', 'addit', 'slam', 'weve', 'lock', 'last', 'month', 'see', 'anoth', 'burn', 'fest'

#### 4.2.6. Stemming

Tahap *stemming* adalah proses mentransformasikan kata majemuk atau kata berafiks menjadi kata dasar. Dalam penelitian ini, *stemming* tidak digunakan pada keseluruhan kata, melainkan pada saat pelabelan. Fungsi ini dibangun ke dalam sintaks / kode pelabelan. Jika suatu kata dalam ulasan tidak dikenali oleh kamus, kata itu akan diubah menjadi bentuk kata dasar, yang kemudian akan dicocokkan dengan kamus. Contoh tersebut dapat dilihat pada tabel 6.

**Tabel 6. Proses *stemming* data**

No	Sebelum	Sesudah
1	'price', 'updat', 'usd', 'follow', 'recent', 'eth', 'price', 'updat'	'price', 'updat', 'usd', 'follow', 'for', 'recent', 'eth', 'price', 'updat'
2	'well', 'understand', 'everi', 'dip', 'price', 'provid', 'golden', 'opportun', 'accumul', 'one', 'understand', 'thi', 'youll', 'know', 'buyhodlb', 'patient', 'follow', 'alway', 'avail', 'dure', 'downher', 'tweetstapredict', 'amp', 'tip', 'accur'	'do', 'well', 'to', 'understand', 'that', 'everi', 'dip', 'in', 'the', 'price', 'of', 'provid', 'golden', 'opportun', 'to', 'accumul', 'one', 'you', 'understand', 'thi', 'youll', 'know', 'to', 'do', 'buyhodlb', 'patient', 'follow', 'she', 'alway', 'avail', 'dure', 'up', 'and', 'downher', 'tweetstapredict', 'amp', 'tip', 'are', 'so', 'accur'
3	'kayla', 'harold', 'egbert', 'laverna'	'kayla', 'harold', 'egbert', 'laverna'
4	'yesterday', 'made', 'slam', 'buyback', 'lock', 'forev', 'addit', 'slam', 'weve', 'lock', 'last', 'month', 'see', 'anoth', 'burn', 'fest'	'yesterday', 'we', 'made', 'slam', 'buyback', 'and', 'lock', 'it', 'forev', 'in', 'addit', 'to', 'the', 'slam', 'weve', 'lock', 'in', 'the', 'last', 'month', 'see', 'you', 'at', 'anoth', 'burn', 'fest'

#### 4.2.7. Pelabelan

Pelabelan dimaksudkan untuk membantu dalam klasifikasi. Klasifikasi yang dilakukan selama proses dimasukkan ke dalam pembelajaran yang diawasi. Akibatnya, sebelum masuk ke tahap klasifikasi, data *review* harus diberi label. Prosedur pelabelan dibagi menjadi tiga tahap. Tahap pertama adalah pelabelan, yang melibatkan perubahan peringkat pengguna menjadi sentimen positif, netral, negatif, hampir positif, dan hampir negatif. Modifikasi ditentukan oleh klasifikasi. Label data dari hasil pelabelan dapat dilihat pada gambar 13 dibawah ini.

id	text	sentimen	polar_sentimen	Subjectivity	text_tokens	stem_text	text_tokens_WSW	text_tokens_fdist
0	price update usd follow for recent eth price u...	0	0.000000	0.250000	[price, update, usd, follow, for, recent, eth, ...	[price, updat, usd, follow, for, recent, eth, ...	{price: 2, 'usd': 1, 'follow': 1, ...}	
1	do well to understand that every dip in the pr...	1	0.366667	0.511111	[do, well, to, understand, that, every, dip, i, ...	[do, well, to, understand, that, everi, dip, i, ...	{'well': 1, 'understand': 2, 'everi': 1, 'dip': ...}	
2	kayla harold egbert lavema	0	0.000000	0.000000	[kayla, harold, egbert, lavema]	[kayla, harold, egbert, lavema]	{'kayla': 1, 'harold': 1, 'egbert': 1, 'lavema': ...}	
3	817coin	0	0.000000	0.000000	{}	{}	{}	
4	yesterday we made slam buyback and locked it ...	0	0.000000	0.066667	[yesterday, we, made, slam, buyback, and, lock...	[yesterday, we, made, slam, buyback, and, lock...	{yesterday': 1, 'made': 1, 'slam': 2, 'buybac': ...}	
...	...	...	...	...	...	...	...	
9995	digital rupee or erupee is form of digital to...	1	0.033333	0.054167	[digital, rupee, or, erupee, is, form, of, dig...	[digit, rupe, or, erupe, is, form, of, digit, ...	{'digit': 3, 'rupe': 2, 'erupe': 1, 'form': 1, ...}	
9996	in bear market short every pump dca every up ...	-1	-0.016667	0.352381	[in, bear, market, short, every, pump, dca, ev...	[in, bear, market, short, everi, pump, dca, ev...	{'bear': 1, 'market': 2, 'short': 1, 'everi': ...}	

**Gambar 13. Proses Label Data Sentimen**

#### 4.2.8. Pembagian Data Latih dan Data Uji

Pelabelan data, bagilah data menjadi data pelatihan dan data pengujian. Karena algoritma klasifikasi yang digunakan adalah pembelajaran terawasi, hal ini diperlukan. Sehingga dibutuhkan data untuk membuat model pembelajaran yang kemudian dievaluasi menggunakan data uji. Pada penelitian ini distribusi data dilakukan berdasarkan beberapa skenario, dengan pembagian data menjadi 80%:20%, 75%:25%, 70%:30%, dan 65%:35%. Hasilnya, jumlah data latih yang digunakan seperti terlihat pada Tabel 7.

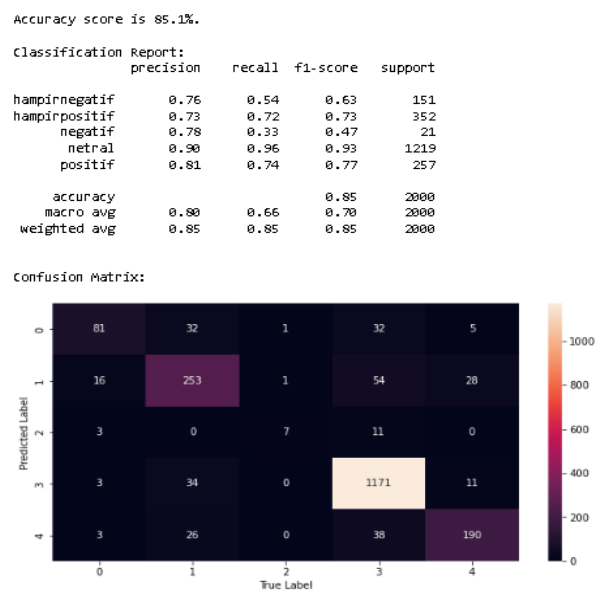
**Tabel 7. Pembagian Data**

Skenario	Pembagian Data		Total
	Data Latih	Data Uji	
Skenario 1 (80:20)	8.000	2.000	10.000
Skenario 2 (75:25)	7.500	2.500	10.000

Skenario 3 (70:30)	7.000	3.000	10.000
Skenario 4 (65:35)	6.500	3.500	10.000

#### 4.2.9. Klasifikasi Support Vector Machine

Klasifikasi algoritma Support Vector Machine(SVM) yang digunakan dalam penelitian ini adalah klasifikasi linier. Klasifikasi yang digunakan menggunakan kernel linier untuk menampilkan hasil *accuracy* 85.1%, *precision* 0.85,*recall* 0.85,*f1-score* 0.85 dan *support* 2000, untuk *precision,recall* dan *f1-score* yang digunakan adalah *weighted average* karena metode ini memperhitungkan ketidakseimbangan dalam jumlah data yang diklasifikasikan dalam setiap kelas, dan kelas yang memiliki banyak data memiliki pengaruh yang lebih besar. Berikut hasil visualisasinya yang berada pada gambar 14 di bawah ini .



Gambar 14. Hasil Visualisasi Confusion Matrix Kernel Linear (80:20)

#### 4.2.10. Klasifikasi Asosiasi

Klasifikasi dalam tahap ini menggunakan analisis asosiasi rules. Analisa tersebut menghasilkan beberapa output untuk menampilkan akurasi dari codingan yang telah di buat. *Output* atau hasil dari codingan tersebut berupa akurasi dari sentimen positif, negatif,netral,hampir positif dan hampir negatif yang memiliki akurasi data latih dan uji sebesar 20 % untuk semua skenario data pada lampiran 47 dan 48. Berikut beberapa contoh *output* dari hasil analisis tersebut untuk data latih apriori yang memiliki akurasi data latih positif(13.45%),negatif(1.26%),netral(61.34%),hampir positif(16.46%),dan hampir negatif (7.49%).

```

In [10]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen netral asosiasi rules: %.2f%%" % (akurasi*100))

Akurasi data latih sentimen netral asosiasi rules: 61.34%

In [12]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen positif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data latih sentimen positif asosiasi rules : 13.45%

In [14]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen negatif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data latih sentimen negatif asosiasi rules : 1.26%

In [16]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen hampir positif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data latih sentimen hampir positif asosiasi rules : 16.46%

In [18]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen hampir negatif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data latih sentimen hampir negatif asosiasi rules : 7.4%

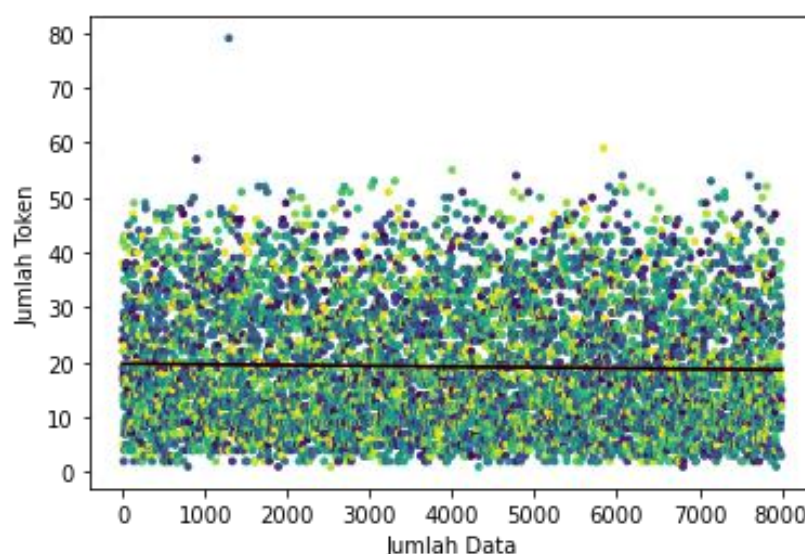
```

**Gambar 15. Hasil Akurasi Data Latih (80:20) Klasifikasi Apriori**

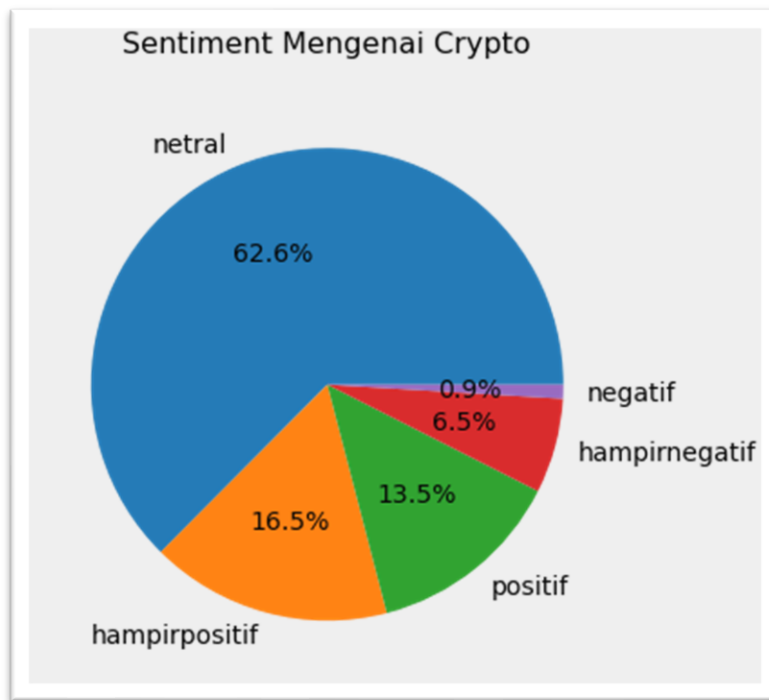
### 4.3 Analisis Hasil dan Evaluasi

#### 4.3.1. Visualisasi Data

Visualisasi menggunakan *bar chart* dan *word cloud* dilakukan untuk menarik secara visual dan membantu analisis sentimen pengguna aplikasi. Tahap asosiasi dilakukan pada sentimen negatif untuk identifikasi kata sering muncul. Regresi pada kernel linear SVM menggunakan scatter diagram untuk menentukan hyperplane dari analisa sentimen dan memisahkan data menjadi dua bagian (nilai batas atas dan bawah), dengan nilai batas atas dan batas bawah terletak pada jumlah token 20, dan grafik pie chart dari data sentimen skenario 80:20 yang memiliki data sentimen positif sebesar 13,5%, negatif sebesar 0,9%, netral sebesar 62,6%, hampir positif sebesar 16,5% dan hampir negatif sebesar 6,5%, visualisasi yang dilakukan seperti gambar 16 dibawah ini.



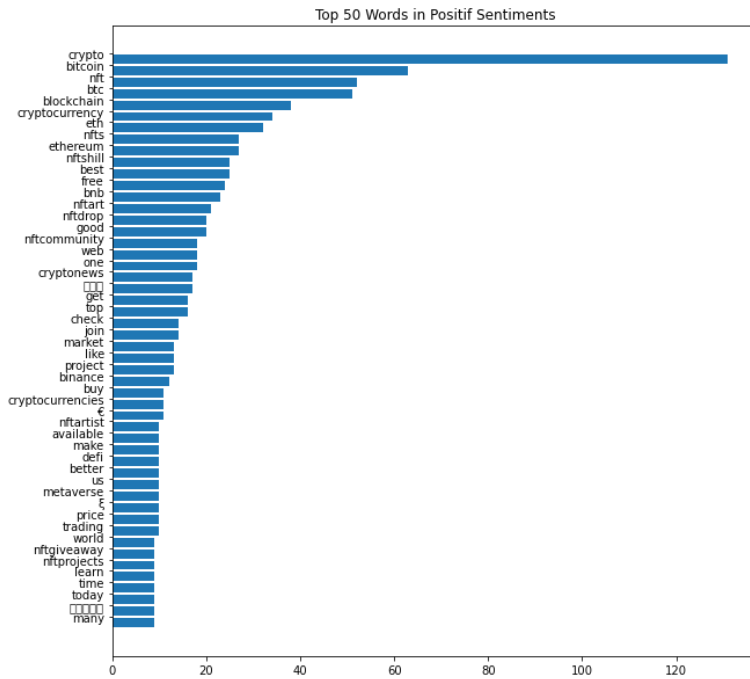
**Gambar 16. Diagram Scatter Plot dan Garis Hyperplane Skenario Data (80:20)**



**Gambar 17. Grafik Pie Chart Sentimen Skenario 80:20**

#### **4.3.2. Sentimen Positif**

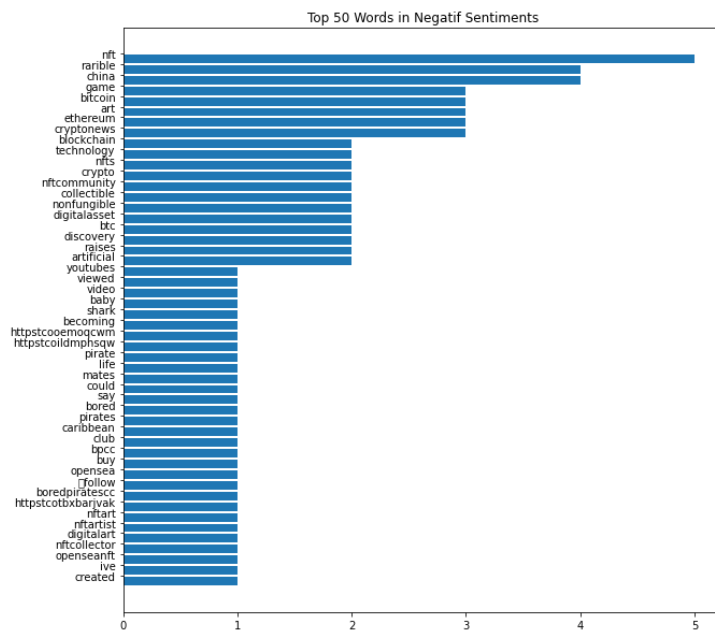
Gambar 18 menunjukkan visualisasi diagram batang sebaran kata sentimen positif dengan tiga kata teratas yaitu *crypto*, *bitcoin* dan *nft*. Tujuannya adalah untuk mempermudah memahami persebaran dan pola kata-kata yang muncul dalam sentimen positif.



Gambar 18. Sebaran Kata Sentimen Positif (80:20)

### 4.3.3. Sentimen Negatif

Gambar 19 menggunakan diagram batang untuk menunjukkan frekuensi kelas kata sentimen positif dengan tiga kata teratas yaitu *nft*, *rarible* dan *china*. Tujuannya adalah untuk mempermudah memahami persebaran dan pola kata-kata yang muncul dalam sentimen negatif.

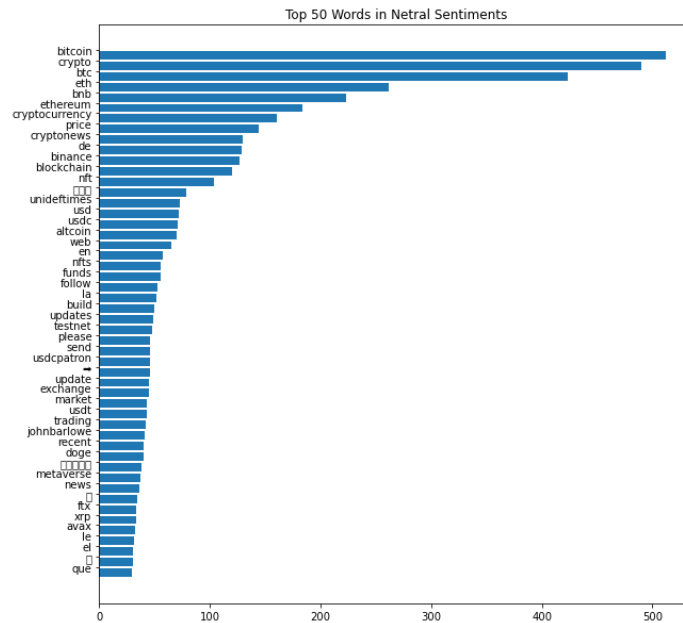




**Gambar 19. Sebaran Kata Sentimen Negatif (80:20)**

#### 4.3.4. Sentimen Netral

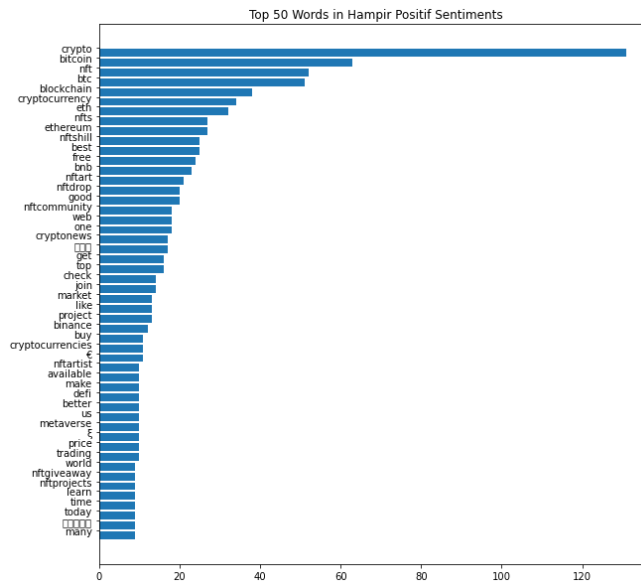
Gambar 20 menggunakan diagram batang untuk menunjukkan frekuensi kelas kata sentimen positif dengan tiga kata teratas yaitu *bitcoin*, *crypto* dan *btc*. Tujuannya adalah untuk mempermudah memahami persebaran dan pola kata-kata yang muncul dalam sentimen negatif.



**Gambar 20. Sebaran Kata Sentimen Netral (80:20)**

#### 4.3.5. Sentimen Hampir Positif

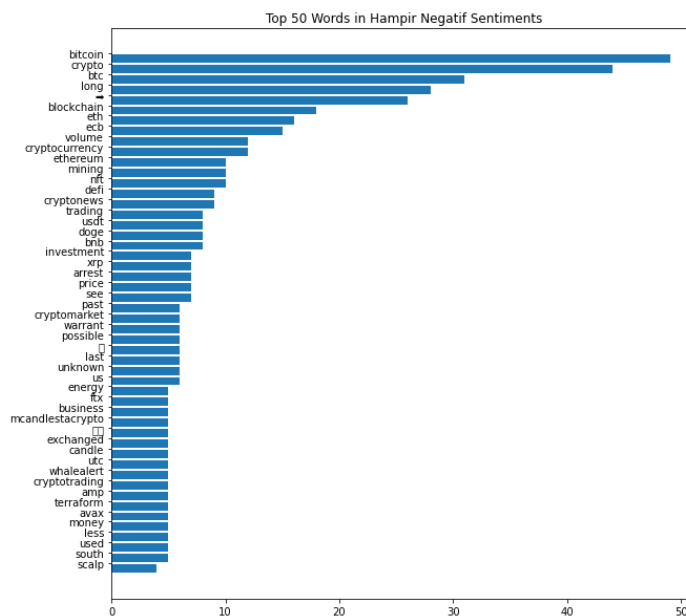
Gambar 21 menggunakan diagram batang untuk menunjukkan frekuensi kelas kata sentimen positif dengan tiga kata teratas yaitu *crypto*, *bitcoin* dan *nft*. Tujuannya adalah untuk mempermudah memahami persebaran dan pola kata-kata yang muncul dalam sentimen hampir positif.



**Gambar 21. Sebaran Kata Sentimen Hampir Positif (80:20)**

#### 4.3.6. Sentimen Hampir Negatif

Gambar 22 menggunakan diagram batang untuk menunjukkan frekuensi kelas kata sentimen positif dengan tiga kata teratas yaitu *bitcoin*, *crypto* dan *btc*. Tujuannya adalah untuk mempermudah memahami persebaran dan pola kata-kata yang muncul dalam sentimen hampir negatif.



**Gambar 22. Sebaran Kata Sentimen Hampir Negatif (80:20)**

#### **4.3.7. Evaluasi**

Hasil analisis yang sudah dilakukan, dapat dilakukan evaluasi yakni, jumlah sebaran data uji dapat menggunakan data yang lebih sedikit dengan skenario percobaan lebih sedikit untuk meningkatkan hasil tingkat akurasi data sentimen. Klasifikasi dalam algoritma *support vector machines* memiliki akurasi lebih tinggi pada data yang lebih sedikit jumlahnya karena memerlukan waktu yang lebih lama untuk data yang berjumlah banyak. Algoritma apriori dalam penelitian memiliki akurasi yang lebih rendah dalam penelitian ini namun memiliki proses klasifikasi yang lebih cepat dibandingkan algoritma SVM karena klasifikasi apriori memiliki proses klasifikasi yang cepat untuk data yang besar.

## **BAB V**

### **HASIL DAN PEMBAHASAN**

#### **5.1 Hasil**

Salah satu metode analisis data adalah analisis deskriptif. Analisis ini menggunakan metode numerik dan grafis untuk mencari pola data, meringkas informasi dalam kumpulan data, dan menyampaikan informasi dalam format yang diinginkan (Santosa R. G., 2004). Menurut Sugiyono (2009), analisis deskriptif adalah penggunaan statistika untuk menganalisis data dengan cara mendeskripsikan atau mendeskripsikan data yang telah dikumpulkan sebagaimana adanya tanpa maksud menarik kesimpulan yang luas atau membuat generalisasi yang luas. Pada penelitian ini, diketahui dari hasil awal proses pencarian data yang dilakukan dari bulan september hingga januari didapatkan jumlah data sebesar 10.000 komentar. Jumlah ulasan tersebut menunjukkan pada bulan November 2021 hingga November 2022 pengguna aplikasi twitter memberikan *feedback* (umpan balik), dengan lokasi geografis *random*.

#### **5.2. Pembahasan**

##### **5.2.1 Klasifikasi SVM**

Klasifikasi SVM pada penelitian menggunakan tipe klasifikasi linier untuk mengolah data sentimen, library pada klasifikasi data yang digunakan adalah *library Tfidf Vectorizer* untuk mengetahui data latih dan data uji pada vektor, *library Linear SVC* untuk mengetahui nilai hasil *precision, recall, f1-score*, dan *support* dan akurasi tertinggi yang diperoleh pada proses pengolahan data uji adalah pada skenario data ketiga dari empat skenario data yaitu 70:30. Data latih yang digunakan berjumlah 7000 data dan data uji yang digunakan berjumlah 3000 data, dengan akurasi data uji sebesar 95% dan akurasi data latih sebesar 79%. Skenario data pertama menghasilkan akurasi data uji sebesar 96%, dan data latih sebesar 75%. Skenario data kedua menghasilkan akurasi data uji sebesar 95%, dan data latih sebesar 76%. Skenario data keempat menghasilkan akurasi data uji sebesar 95%, dan data latih sebesar 78%. Pengolahan keempat data skenario pada data skenario pertama menghasilkan akurasi 95% lalu data skenario kedua 95%, data skenario ketiga 95%, dan data skenario keempat 95%, dapat dilihat bahwa akurasi tidak mengalami penurunan dan kenaikan pada skenario pertama hingga keempat pada data uji dan kejadian penurunan dan kenaikan pada skenario pertama hingga keempat pada komposisi data latih yang digunakan dalam skenario, kenaikan terjadi pada komposisi data 80:20(75%), 75:25(76%), 70:30(79%), lalu setelahnya mengalami penurunan di komposisi data 65:35(78%). Klasifikasi sentimen emotikon pada skenario pertama data uji sebesar 94% , data latih sebesar 72%, skenario kedua data uji sebesar 92%, data latih sebesar 71%, skenario ketiga data uji sebesar 93%, data latih sebesar 72%, skenario keempat data uji 93%, dan data latih sebesar 72%.

```
In [9]: import numpy as np
from sklearn.svm import SVC

#lin_svm=svm.LinearSVC()
lin_svm= svm.SVC(kernel='linear',C=1)
lin_svm.fit(train_vectors, data_train['sentimen'])
lin_svm.fit(test_vectors, data_test['sentimen'])
y_train_hat=lin_svm.predict(train_vectors)
y_test_hat=lin_svm.predict(test_vectors)

print("train accuracy= %.0f%%" % (np.mean(y_train_hat == data_train['sentimen']) * 100))
print("test accuracy= %.0f%%" % (np.mean(y_test_hat == data_test['sentimen']) * 100))
```

train accuracy= 74%  
test accuracy= 95%

**Gambar 23. Hasil Penghitungan Akurasi Sentimen Klasifikasi SVM (80:20)**

```
In [13]: import numpy as np
from sklearn.svm import SVC

#lin_svm=svm.LinearSVC()
lin_svm= svm.SVC(kernel='linear',C=1)
lin_svm.fit(train_vectors, data_train['sentimen_emot'])
lin_svm.fit(test_vectors, data_test['sentimen_emot'])
y_train_hat=lin_svm.predict(train_vectors)
y_test_hat=lin_svm.predict(test_vectors)

print("train accuracy_analisa emoticon= %.0f%%" % (np.mean(y_train_hat == data_train['sentimen_emot']) * 100))
print("test accuracy_analisa emoticon= %.0f%%" % (np.mean(y_test_hat == data_test['sentimen_emot']) * 100))
```

train accuracy\_analisa emoticon= 69%  
test accuracy\_analisa emoticon= 94%

**Gambar 24. Hasil Penghitungan Akurasi Emotikon Sentimen Klasifikasi SVM (80:20)**

### 5.2.2 Klasifikasi Asosiasi Rules dan Visualisasi

Klasifikasi Asosiasi Rules pada penelitian ini menggunakan parameter support dan confidence dari setiap transaksi atau dalam kata lain baris data. *Library* yang digunakan *Library* Apriori merupakan salah satu *library* python yang digunakan untuk melakukan analisis association rule pada data *mining*. *Library* ini digunakan untuk menemukan pola asosiasi atau korelasi antara berbagai item pada *dataset* yang besar. *Library* matplotlib.pyplot adalah salah satu *library* dari Matplotlib yang digunakan untuk membuat visualisasi data berupa plot atau grafik pada bahasa pemrograman Python. *Library* ini sangat berguna dalam melakukan eksplorasi data dan juga visualisasi data secara interaktif. Nilai *support* yang ditentukan dalam penelitian ini sebesar 0,1 dan penentuan nilai confidence dalam penelitian ini sebesar 0,8. Penentuan nilai *support* dan *confidence* dilihat berdasarkan jumlah kemunculan kata positif, negatif, netral, hampir positif dan hampir negatif. Data kemunculan kata positif memiliki kemunculan kata sekitar 120 kata, kemunculan kata negatif sampai 5 kata, kata netral 500 kata, hampir positif 120 kata, hampir negatif 50 kata sehingga dalam penelitian ini menggunakan nilai support paling rendah yaitu 0.1 untuk memperoleh hasil yang lebih akurat. Akurasi data uji sentimen pada skenario pertama 80:20 adalah netral sebesar 6,35 %, positif sebesar 6,35 %, negatif sebesar 6,35%, hampir positif sebesar 6,35%, dan hampir negatif sebesar 6,35%. Akurasi data latih pada sentimen pada skenario pertama 80:20 adalah netral sebesar 7,49 %, positif sebesar 7,49%, negatif sebesar 7,49%, hampir positif sebesar 7,49%, dan hampir negatif sebesar 7,49%. Akurasi data

uji sentimen pada skenario kedua 75:25 adalah netral sebesar 6,20 %, positif sebesar 6,20% negatif sebesar 6,20% hampir positif sebesar 6,20%, dan hampir negatif sebesar 6,20%. Akurasi data latih sentimen pada skenario kedua 75:25 adalah netral sebesar 7,61%, positif sebesar 7,61%,negatif sebesar 7,61%. Akurasi data uji sentimen pada skenario ketiga 70:30 adalah netral sebesar 6,50 %, positif sebesar 6,50 %,negatif sebesar 6,50%. Akurasi data latih sentimen pada skenario ketiga 70:30 adalah netral sebesar 7,59%, positif sebesar 7,59%,negatif sebesar 7,59%,da hampir positif sebesar 7,59%, dan hampir negatif sebesar 7,59%. Akurasi data uji sentimen pada skenario keempat 65:35 adalah netral sebesar 6,91%, positif sebesar 6,91%,negatif sebesar 6,91%,hampir positif sebesar 6,91%, dan hampir negatif sebesar 6,91%. Akurasi data latih sentimen pada skenario keempat 65:35 adalah netral sebesar 7,45 %, positif 7,45%,negatif sebesar 7,45%, hampir positif sebesar 7,45%, dan hampir negatif sebesar 7,45%. Akurasi dari semua skenario data latih dan data uji sama, yaitu sebesar 20%.

```

akurasi_latih_netral = np.mean(prediksi_netral == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_netral)

# Menghitung akurasi data latih positif
akurasi_latih_positif = np.mean(prediksi_positif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_positif)

# Menghitung akurasi data latih negatif
akurasi_latih_negatif = np.mean(prediksi_negatif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_negatif)

# Menghitung akurasi data latih hampir positif
akurasi_latih_hampirpositif = np.mean(prediksi_hampirpositif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_hampirpositif)

# Menghitung akurasi data latih hampir negatif
akurasi_latih_hampirnegatif = np.mean(prediksi_hampirnegatif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_hampirnegatif)

# Menghitung rata-rata total akurasi data latih
rata_rata_akurasi_latih = np.mean(akurasi_latih)

# Menampilkan hasil
print("Rata-rata akurasi data latih: %.2f%%" % (rata_rata_akurasi_latih * 100))

```

Rata-rata akurasi data latih: 20.00%

**Gambar 25. Akurasi Data Latih Sentimen Klasifikasi Apriori (80:20)**

```

if set(itemset.items).issubset(komentar):
    sentimen = itemset.support
    break
    prediksi_hampirnegatif.append(sentimen)

# Menghitung akurasi data latih netral
akurasi_uji_netral = np.mean(prediksi_netral == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_netral)

# Menghitung akurasi data latih positif
akurasi_uji_positif = np.mean(prediksi_positif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_positif)

# Menghitung akurasi data latih negatif
akurasi_uji_negatif = np.mean(prediksi_negatif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_negatif)

# Menghitung akurasi data latih hampir positif
akurasi_uji_hampirpositif = np.mean(prediksi_hampirpositif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_hampirpositif)

# Menghitung akurasi data latih hampir negatif
akurasi_uji_hampirnegatif = np.mean(prediksi_hampirnegatif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_hampirnegatif)

# Menghitung rata-rata total akurasi data latih
rata_rata_akurasi_uji = np.mean(akurasi_uji)

# Menampilkan hasil
print("Rata-rata akurasi data uji: %.2f%%" % (rata_rata_akurasi_uji * 100))

```

Rata-rata akurasi data uji: 20.00%

**Gambar 26. Akurasi Data Uji Sentimen Klasifikasi Apriori (80:20)**

## **BAB VI PENUTUP**

### **6.1 Kesimpulan**

Hasil dari pengolahan data, dan hasil dari pembahasan penelitian ini adalah sebagai berikut :

1. Klasifikasi *support vector machines* memiliki tingkat akurasi analisis data sentimen yang lebih tinggi dengan persentase data latih tertinggi sebesar 79% dalam skenario data ketiga yaitu 70:30. Parameter ini membuktikan bahwa analisa menggunakan algoritma *support vector machines* sangat cocok untuk analisis sentimen sedangkan untuk analisa menggunakan klasifikasi asosiasi atau apriori memiliki akurasi prediksi data latih tertingginya dari semua skenario sebesar 20%.

2. Visualisasi menggunakan *word cloud* dan *bar chart* dalam proses pengolahan data menggunakan klasifikasi *support vector machines* dapat memudahkan peneliti untuk meneliti kata apa saja yang sering muncul yaitu sentimen positif ‘crypto’ (200 kata), netral ‘prince’ (470 kata), negatif ‘usdt’(90 kata), informasi ini berguna bagi pengamat ekonomi untuk melihat komentar yang sering muncul dalam topik crypto bitcoin, untuk mengambil suatu keputusan dalam investasi jangka panjang dan pendek. Klasifikasi ini membentuk pola lurus datar yang memisahkan data token menjadi dua wilayah, wilayah atas untuk kemunculan kata tokenisasi diatas 20 sedangkan wilayah untuk kemunculan kata tokenisasi dibawah 10.

3. Visualisasi *bar chart* pada sentimen positif skenario pertama sebagai contoh, peneliti akan melihat dari 5 kata teratas dari 50 kata teratas dalam visualisasi yakni ‘crypto’ (120 kata), ‘bitcoin’ (60 kata), ‘nft’ (55 kata), ‘btc’(50 kata) dan ‘blockchain’ (40 kata) dapat dilihat pada gambar 18. Visualisasi *bar chart* kata sentimen negatif yakni ‘nft’(5 kata), ‘rarible’ (4 kata), ‘china’ (4 kata), ‘game’ (3 kata), dan ‘bitcoin’ (3 kata) dapat dilihat pada gambar 19. Visualisasi *bar chart* kata sentimen netral yakni ‘bitcoin’ (550 kata), ‘crypto’ (400 kata), ‘btc’ (380 kata), ‘eth’ (350 kata), dan bnb (182 kata) dapat dilihat pada gambar 20. Visualisasi *bar chart* kata sentimen hampir positif yakni ‘crypto’(160 kata), ‘bitcoin’ (45 kata), ‘nft’ (43 kata), ‘blockchain’ (41 kata), dan ‘btc’ (40 kata) dapat dilihat pada gambar 21 Visualisasi *bar chart* kata sentimen hampir negatif yakni ‘btc’(60 kata), ‘crypto’ (58 kata), ‘bitcoin’ (50 kata), ‘usdt’ (25 kata), dan ‘volume’ (20 kata) dapat dilihat pada gambar 22. Data visualisasi kata tertinggi dilihat dari 5 kata yang sering muncul , yakni berada pada kata ‘bitcoin’ (550 kata) dalam sentimen netral , dan kata terendah terdapat pada kata ‘game’ dan ‘bitcoin’(3 kata) pada sentimen negatif . Klasifikasi dengan kernel linear dalam diagram *scatter plot* menghasilkan *hyper plane* pada titik kisaran 10 sampai dengan 20 Jumlah token dengan empat skenario data. Klasifikasi ini membentuk pola lurus datar yang memisahkan data token menjadi dua wilayah, wilayah atas untuk kemunculan kata tokenisasi diatas 20 sedangkan wilayah untuk kemunculan kata tokenisasi dibawah 10.



## 6.2 Saran

Saran untuk penelitian selanjutnya ,berdasarkan penelitian ini yaitu :

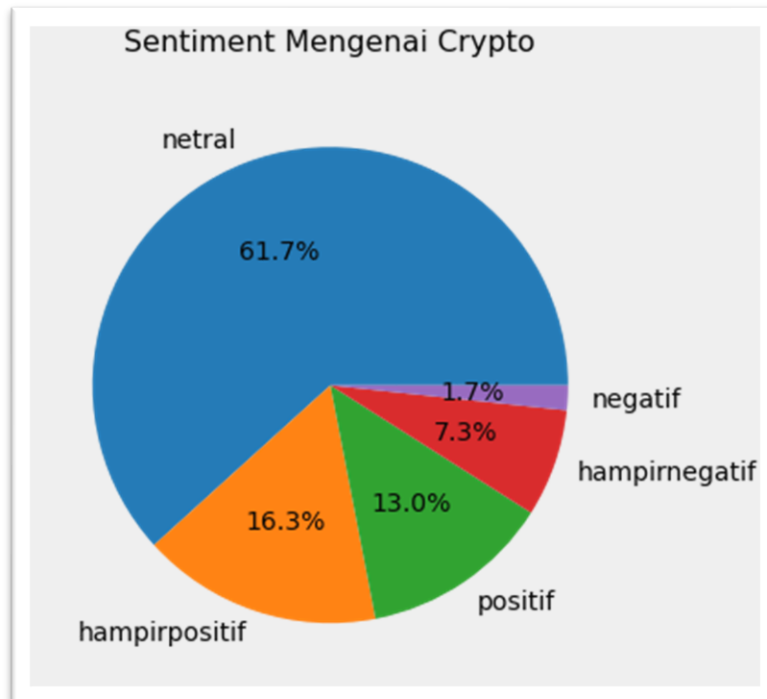
1. Pengklasifikasian data yang lebih simpel dan akurat sehingga dapat memunculkan model *machine learning* yang lebih baik.
2. Penggunaan *tools* analisis sentimen yang lebih baik dari penelitian ini di masa depan yaitu dengan penggunaan framework seperti *google colab* atau kombinasi pengolahan dengan penggunaan *framework react.js* untuk visualisasi datanya.

## DAFTAR PUSTAKA

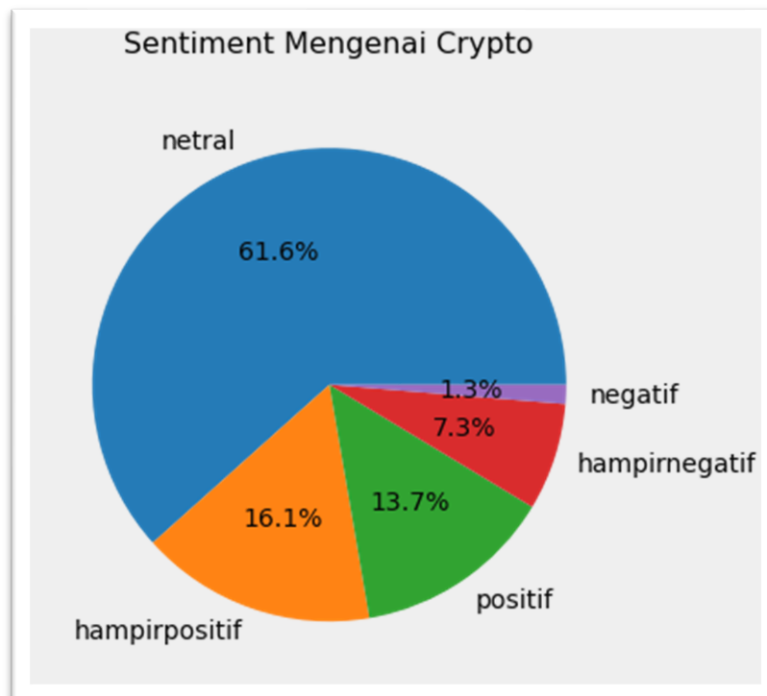
- Samsudiney, "Penjelasan Sederhana Tentang Apa Itu SVM," <https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02>, Jul. 2019.
- R. Tineges, A. Triayudi, and I. D. Sholihati, "Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM)," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 4, no. 3, p. 650, Jul. 2020, doi: 10.30865/mib.v4i3.2181.
- LP2M, "Analisis Sentimen (Sentiment Analysis) : Definisi, Tipe dan Cara Kerjanya," <https://lp2m.uma.ac.id/2022/02/21/analisis-sentimen-sentiment-analysis-definisi-tipe-dan-cara-kerjanya/>, Feb. 21, 2022.
- H. C. Husada and A. S. Paramita, "Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Teknika*, vol. 10, no. 1, pp. 18–26, Feb. 2021, doi: 10.34148/teknika.v10i1.311.
- Angelina Fransiska, "METODE DATA MINING CLASSIFICATION," <https://sis.binus.ac.id/2021/10/22/metode-data-mining-classification/>, Oct. 22, 2021.
- DQLAB, "Kenali Data Science Algoritma Apriori pada Machine Learning," <https://dqlab.id/kenali-data-science-algoritma-apriori-pada-machine-learning#:~:text=sampai%20selesai%20ya!,1.,frequent%20tertinggi%20dari%20suatu%20database.>, Aug. 18, 2022.
- BINUS, "Apa Itu Text Mining? ," <https://sis.binus.ac.id/2021/04/23/apa-itu-text-mining/>, Apr. 23, 2021.
- Rudi Dian Arifin, "Pengertian Twitter beserta Sejarah, Fitur, Fungsi, Manfaat, dll.," <https://dianisa.com/pengertian-twitter/>, Mar. 02, 2023.
- Algoritma, "Mengenal python dan kegunaannya," <https://algorit.ma/blog/data-science/apa-itu-python-2022/>, Apr. 22, 2022.
- A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, "ANALISIS SENTIMEN APLIKASI RUANG GURU DI TWITTER MENGGUNAKAN ALGORITMA KLASIFIKASI," *Jurnal Teknoinfo*, vol. 14, no. 2, p. 115, Jul. 2020, doi: 10.33365/jti.v14i2.679.
- A. D. Prasetyo and T. Informasi, "ALGORITMA APRIORI UNTUK ANALISIS POLA PEMBELIAN OBAT KONSUMEN," *Cyberarea.id*, vol. 2, no. 9, pp. 1–16, 2022.

- Ahmad Muhandian, “Memahami: Apa itu Numpy pada Python?,” <https://www.petanikode.com/python-numpy/>, Nov. 12, 2022.
- Yunita Sari, “Pengenalan Natural Language Toolkit (NLTK) Bagian 1,” Yogyakarta, Sep. 2019.
- Suripto, “Teknik Pre-Processing dan Classification Dalam Data Science,” <https://mie.binus.ac.id/2022/08/26/teknik-pre-processing-dan-classification-dalam-datascience/#:~:text=Pengertian%20lain%20menyebutkan%20bahwa%20data,awal%20sebelum%20melakukan%20data%20mining.>, Aug. 26, 2022.
- Sampoerna University, “Pengertian Rumusan Masalah, Cara Membuat Hingga Contohnya,” <https://www.sampoernauniversity.ac.id/id/rumusan-masalah/>, Mar. 08, 2022.
- Politeknik Indonesia, “Apa Itu Analisis Hasil Penelitian?,” <https://polindo.ac.id/aplicatlon/pendidikan/apa-itu-analisis-hasil-penelitian3505.php>, Nov. 16, 2021.
- Rahmadya Trias Handayanto, “Membagi Data Latih dan Uji Secara Otomatis Pada Python,” <https://rahmadya.com/2020/04/10/membagi-data-latih-dan-uji-secara-otomatis-padapython/#:~:text=Mengapa%20harus%20memisahkan%20data%20latih,digunakan%20dalam%20pelatihan%20sangat%20buruk.>, Apr. 10, 2020.
- Algoritma, “Jupyter: Pengertian,Fitur, Dan Fungsi,” <https://algorit.ma/blog/cara-menggunakan-jupyter-notebook-2022/>, Mar. 15, 2022.
- Restu, “Literature Review : Pengertian, Metode, Manfaat, dan Cara Membuat,” <https://www.gramedia.com/literasi/literature-review/>, 2021.
- Microsoft, “Apa itu klasifikasi data?,” <https://learn.microsoft.com/id-id/azure/cloud-adoption-framework/govern/policy-compliance/data-classification>, Mar. 23, 2023.

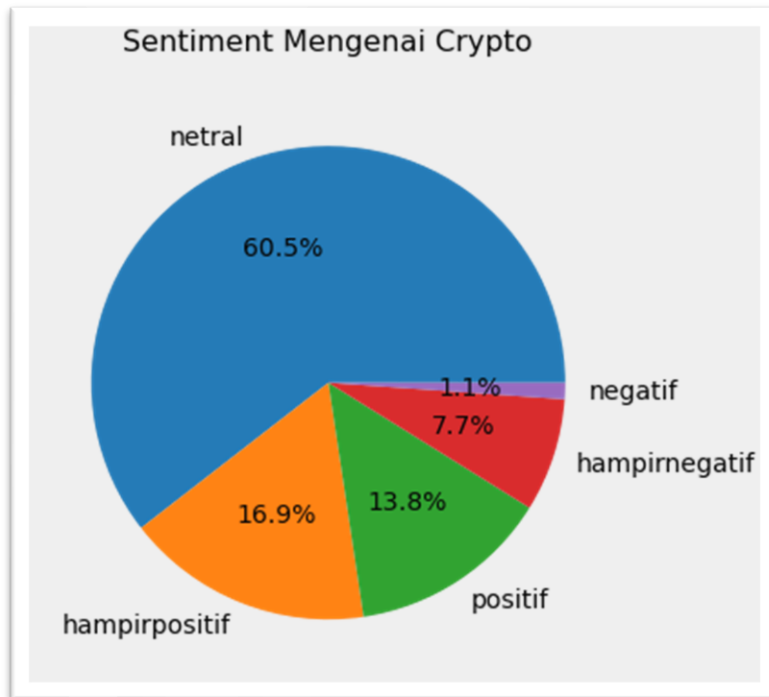
## **LAMPIRAN**



Lampiran 1. Grafik Pie Chart Sentimen Skenario 75:25



Lampiran 2. Grafik Pie Chart Sentimen Skenario 70:30

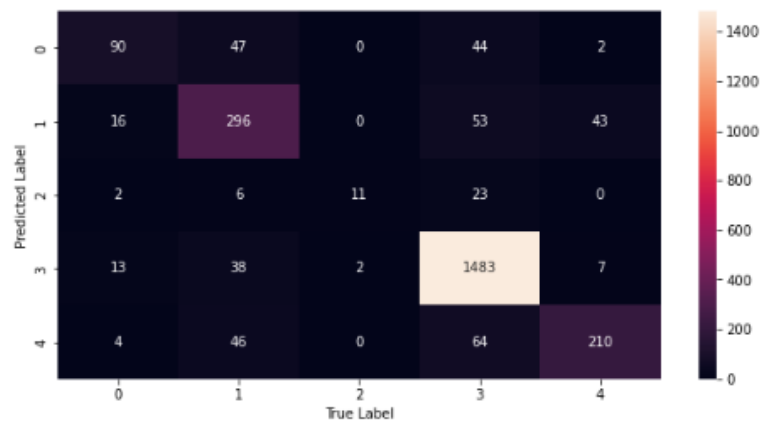


**Lampiran 3. Grafik Pie Chart Sentimen Skenario 65:35**

Accuracy score is 83.6%.

Classification Report:		precision	recall	f1-score	support
hampirnegatif		0.72	0.49	0.58	183
hampirpositif		0.68	0.73	0.70	408
negatif		0.85	0.26	0.40	42
netral		0.89	0.96	0.92	1543
positif		0.80	0.65	0.72	324
accuracy				0.84	2500
macro avg		0.79	0.62	0.67	2500
weighted avg		0.83	0.84	0.83	2500

Confusion Matrix:



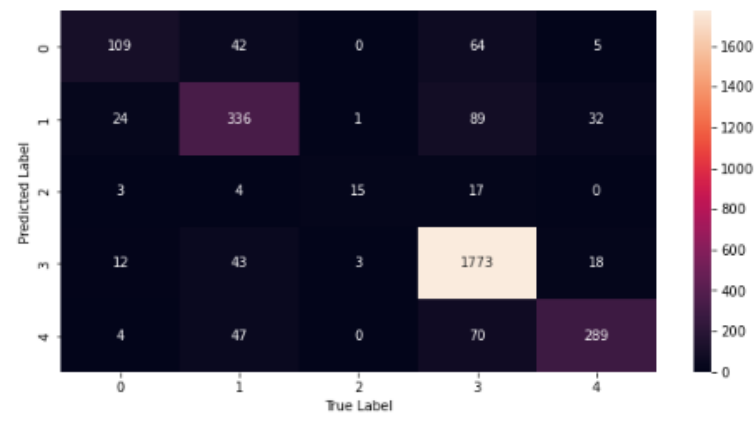
**Lampiran 4. Hasil Visualisasi Confusion Matrix Kernel Linear (75:25)**

Accuracy score is 84.1%.

Classification Report:

	precision	recall	f1-score	support
hampirnegatif	0.72	0.50	0.59	220
hampirpositif	0.71	0.70	0.70	482
negatif	0.79	0.38	0.52	39
netral	0.88	0.96	0.92	1849
positif	0.84	0.70	0.77	410
accuracy			0.84	3000
macro avg	0.79	0.65	0.70	3000
weighted avg	0.83	0.84	0.83	3000

Confusion Matrix:



Lampiran 5. Hasil Visualisasi Confusion Matrix Kernel Linear (70:30)

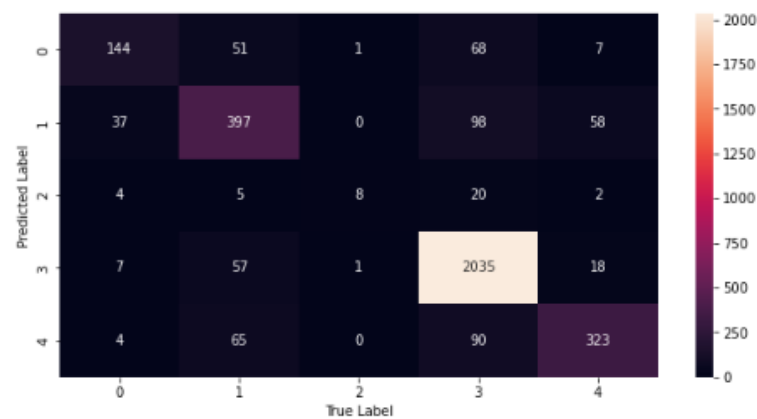


Accuracy score is 83.1%.

Classification Report:

	precision	recall	f1-score	support
hampirnegatif	0.73	0.53	0.62	271
hampirpositif	0.69	0.67	0.68	590
negatif	0.80	0.21	0.33	39
netral	0.88	0.96	0.92	2118
positif	0.79	0.67	0.73	482
accuracy			0.83	3500
macro avg	0.78	0.61	0.65	3500
weighted avg	0.82	0.83	0.82	3500

Confusion Matrix:



## Lampiran 6. Hasil Visualisasi Confusion Matrix Kernel Linear (65:35)

```
In [21]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen netral asosiasi rules : %.2f%%" % (akurasi*100))
Akurasi data uji sentimen netral asosiasi rules : 61.00%
```

```
In [23]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen positif asosiasi rules : %.2f%%" % (akurasi*100))
Akurasi data uji sentimen positif asosiasi rules : 13.50%
```

```
In [25]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen negatif asosiasi rules : %.2f%%" % (akurasi*100))
Akurasi data uji sentimen negatif asosiasi rules : 1.65%
```

```
In [27]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen hampir positif asosiasi rules : %.2f%%" % (akurasi*100))
Akurasi data uji sentimen hampir positif asosiasi rules : 17.50%
```

```
In [29]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen hampir negatif asosiasi rules : %.2f%%" % (akurasi*100))
Akurasi data uji sentimen hampir negatif asosiasi rules : 6.35%
```

## Lampiran 7. Hasil Akurasi Data Uji (80:20) Klasifikasi Apriori

```
In [10]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen netral asosiasi rules: %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen netral asosiasi rules: 61.47%

```
In [12]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen positif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen positif asosiasi rules : 13.32%

```
In [14]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen negatif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen negatif asosiasi rules : 1.21%

```
In [16]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen hampir positif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen hampir positif asosiasi rules : 16.39%

```
In [18]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen hampir negatif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen hampir negatif asosiasi rules : 7.61%

## Lampiran 8. Hasil Akurasi Data Latih (75:25) Klasifikasi Apriori

```
In [21]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen netral asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen netral asosiasi rules : 60.68%

```
In [23]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen positif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen positif asosiasi rules : 13.88%

```
In [25]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen negatif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen negatif asosiasi rules : 1.72%

```
In [27]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen hampir positif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen hampir positif asosiasi rules : 17.52%

```
In [29]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen hampir negatif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen hampir negatif asosiasi rules : 6.20%

## Lampiran 9. Hasil Akurasi Data Uji (75:25) Klasifikasi Apriori

```
In [10]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen netral asosiasi rules: %.2f%%" % (akurasi*100))

Akurasi data latih sentimen netral asosiasi rules: 61.41%
```

```
In [12]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen positif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data latih sentimen positif asosiasi rules : 13.31%
```

```
In [14]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen negatif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data latih sentimen negatif asosiasi rules : 1.19%
```

```
In [16]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen hampir positif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data latih sentimen hampir positif asosiasi rules : 16.50%
```

```
In [18]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen hampir negatif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data latih sentimen hampir negatif asosiasi rules : 7.59%
```

## Lampiran 10. Hasil Akurasi Data Latih (70:30) Klasifikasi Apriori

```
In [21]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen netral asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data uji sentimen netral asosiasi rules : 60.93%
```

```
In [23]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen positif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data uji sentimen positif asosiasi rules : 13.80%
```

```
In [25]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen negatif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data uji sentimen negatif asosiasi rules : 1.70%
```

```
In [27]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen hampir positif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data uji sentimen hampir positif asosiasi rules : 17.07%
```

```
In [29]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen hampir negatif asosiasi rules : %.2f%%" % (akurasi*100))

Akurasi data uji sentimen hampir negatif asosiasi rules : 6.50%
```

## Lampiran 11. Hasil Akurasi Data Uji (70:30) Klasifikasi Apriori

```
In [10]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen netral asosiasi rules: %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen netral asosiasi rules: 61.63%

```
In [12]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen positif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen positif asosiasi rules : 13.43%

```
In [14]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen negatif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen negatif asosiasi rules : 1.09%

```
In [16]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen hampir positif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen hampir positif asosiasi rules : 16.40%

```
In [18]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_latih["sentimen"])
print("Akurasi data latih sentimen hampir negatif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data latih sentimen hampir negatif asosiasi rules : 7.45%

## Lampiran 12. Hasil Akurasi Data Latih (65:35) Klasifikasi Apriori

```
In [21]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen netral asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen netral asosiasi rules : 60.60%

```
In [23]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen positif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen positif asosiasi rules : 13.51%

```
In [25]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen negatif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen negatif asosiasi rules : 1.80%

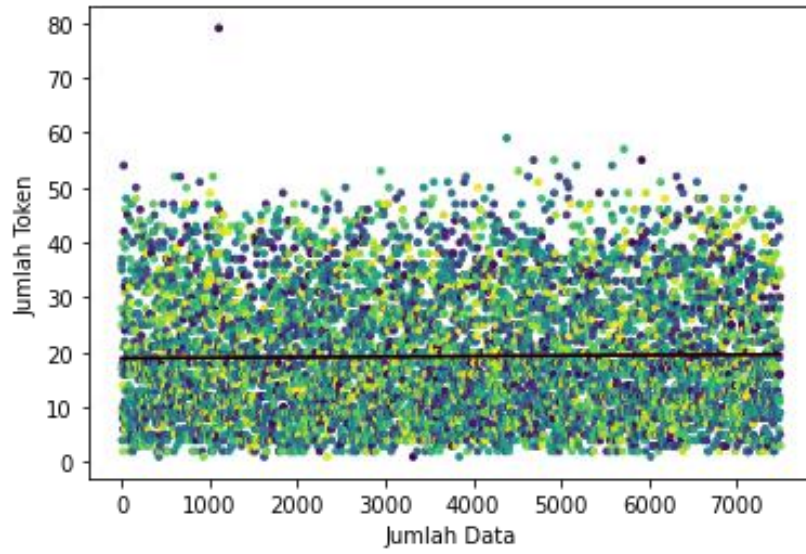
```
In [27]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen hampir positif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen hampir positif asosiasi rules : 17.17%

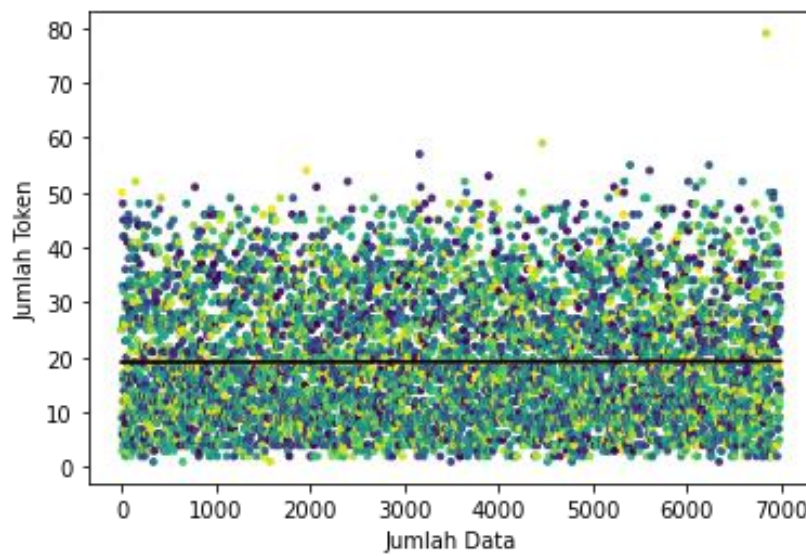
```
In [29]: # Menghitung akurasi dari hasil asosiasi rules
akurasi = np.mean(prediksi == data_uji["sentimen"])
print("Akurasi data uji sentimen hampir negatif asosiasi rules : %.2f%%" % (akurasi*100))
```

Akurasi data uji sentimen hampir negatif asosiasi rules : 6.91%

## Lampiran 13. Hasil Akurasi Data Uji (65:35) Klasifikasi Apriori



**Lampiran 14. Diagram Scatter Plot dan Garis Hyperplane Skenario Data (75:25)**



**Lampiran 15. Diagram Scatter Plot dan Garis Hyperplane Skenario Data (70:30)**

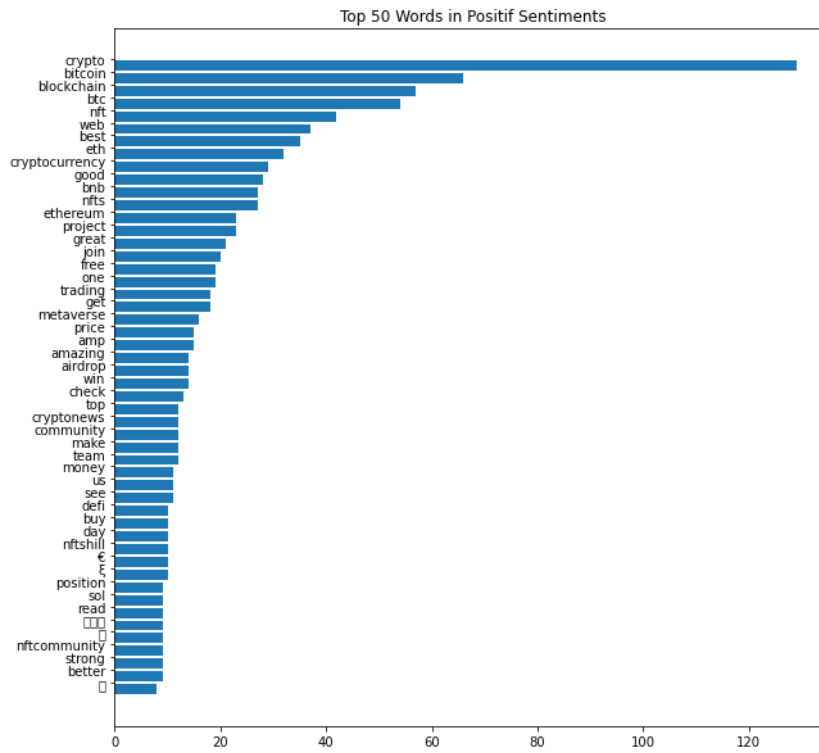








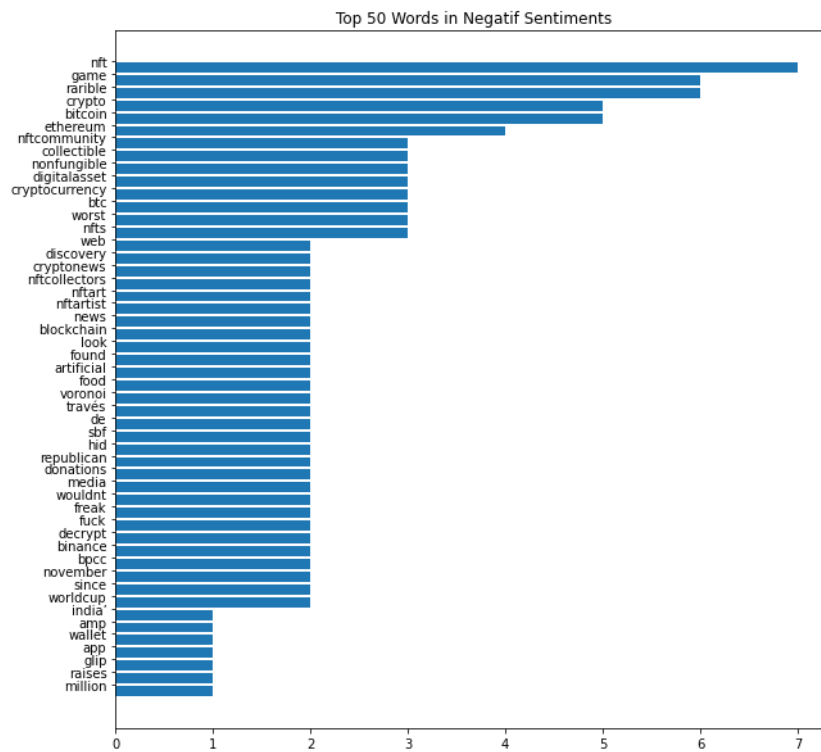




Lampiran 22. Sebaran Kata Sentimen Positif (75:25)



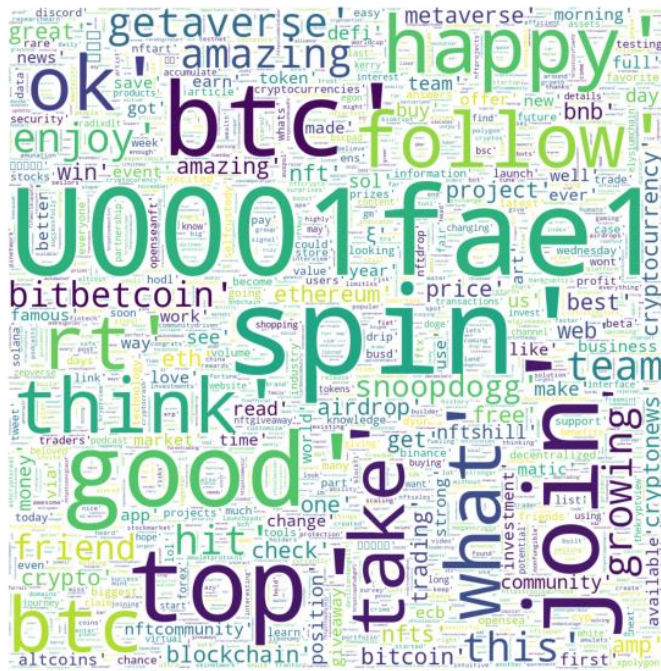
### Lampiran 23. Sebaran Kata Word Cloud Sentimen Positif (75:25)



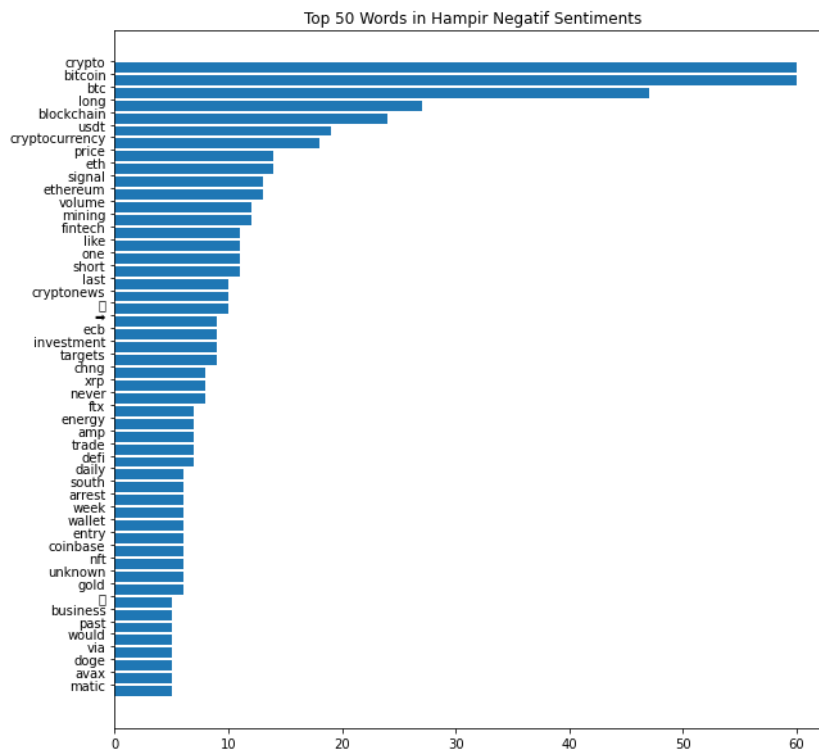
### Lampiran 24. Sebaran Kata Sentimen Negatif (75:25)







Lampiran 29. Sebaran Kata Wordcloud Sentimen Hampir Positif (75:25)



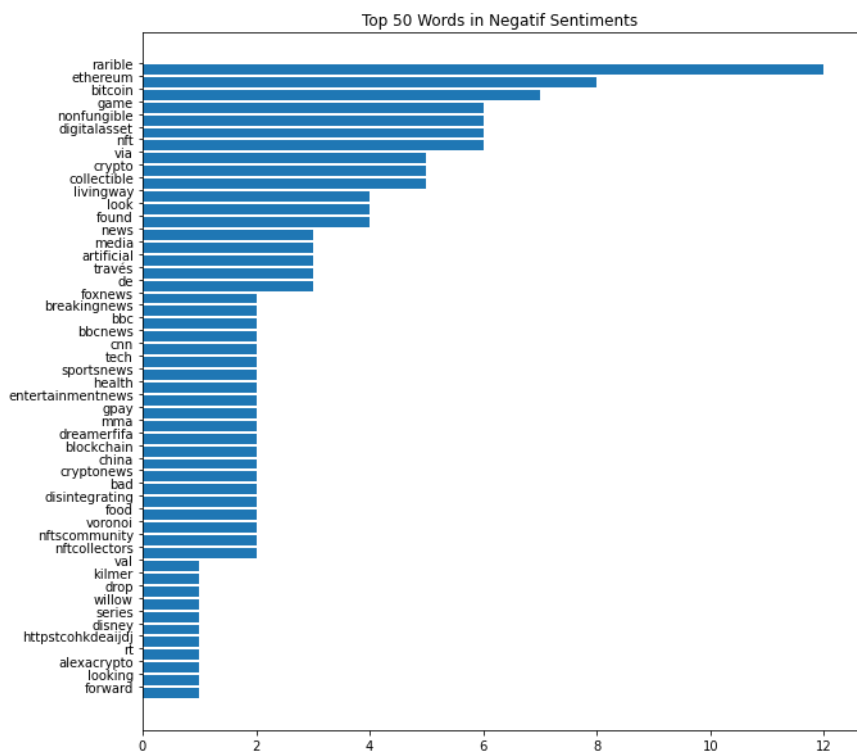
Lampiran 30. Sebaran Kata Sentimen Hampir Negatif (75:25)







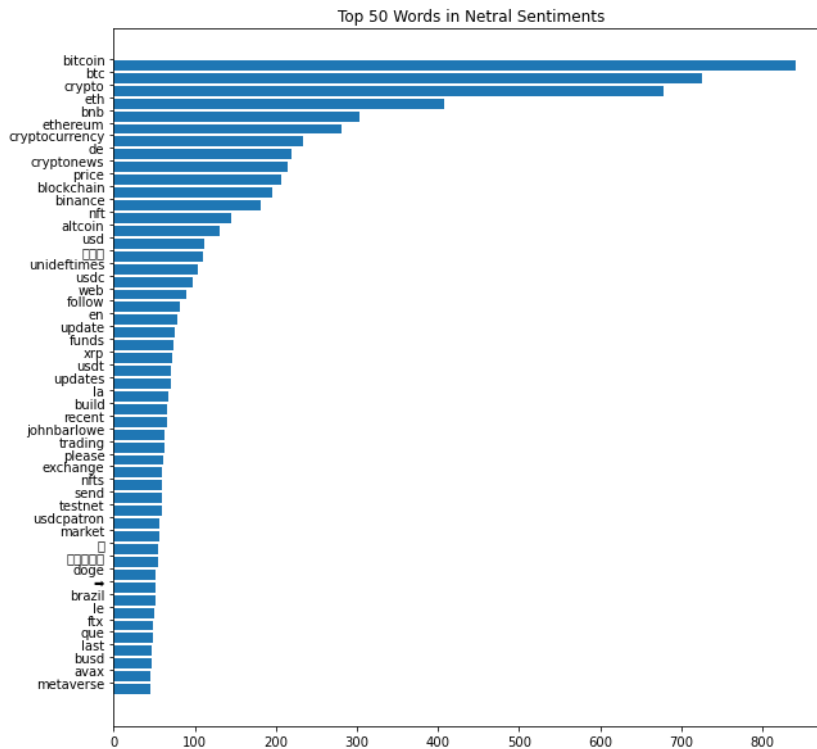
Lampiran 33. Sebaran Kata Wordcloud Sentimen Positif (70:30)



Lampiran 34. Sebaran Kata Sentimen Negatif (70:30)



Lampiran 35. Sebaran Kata Wordcloud Sentimen Negatif (70:30)



Lampiran 36. Sebaran Kata Sentimen Netral (70:30)







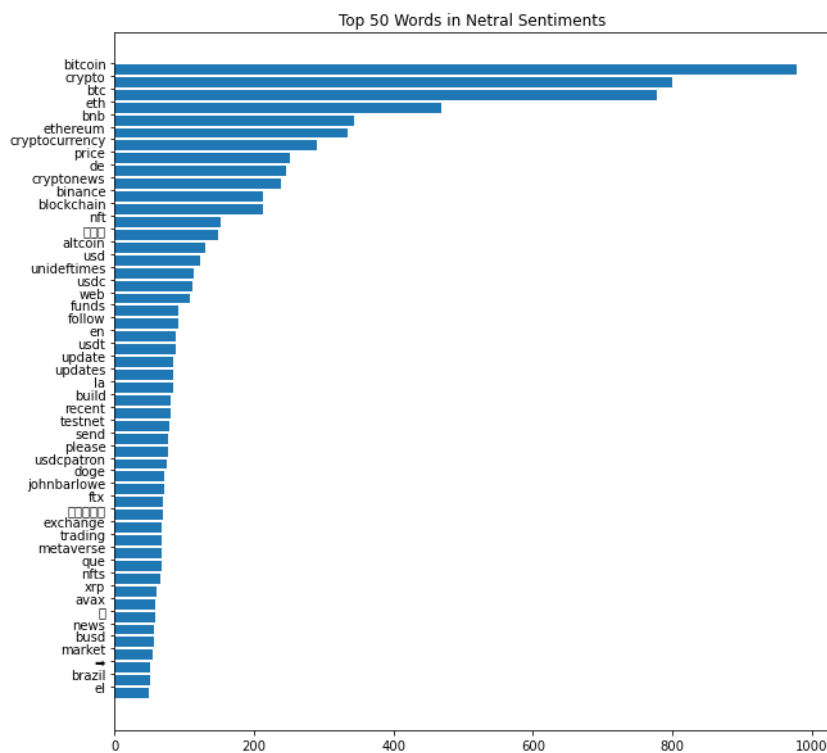




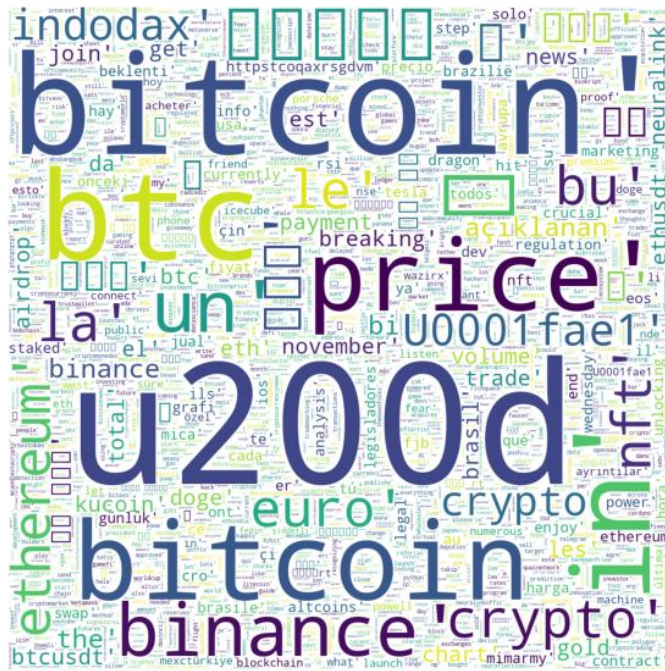




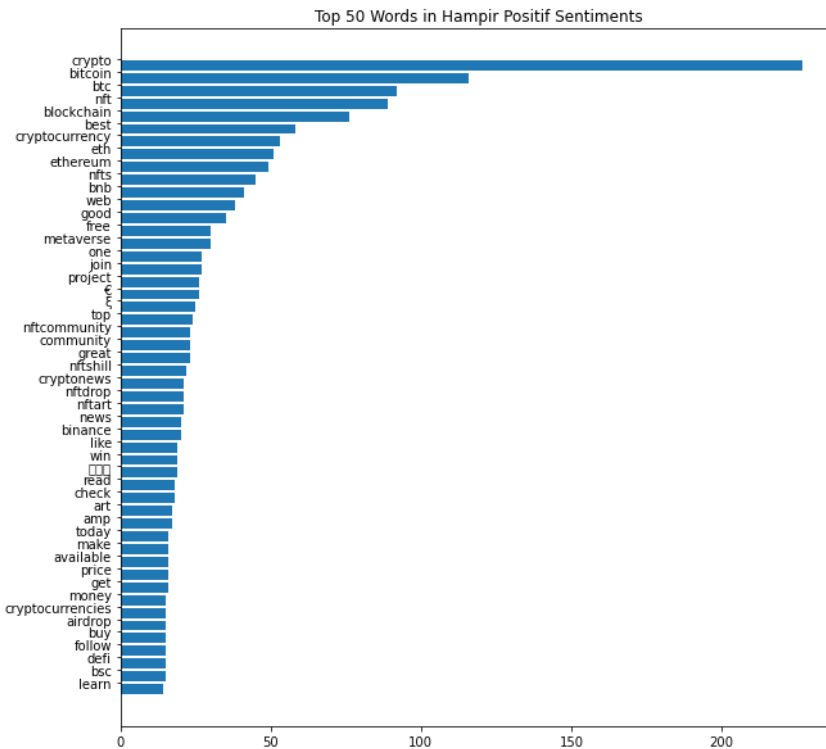
Lampiran 45. Sebaran Kata Wordcloud Sentimen Negatif (65:35)



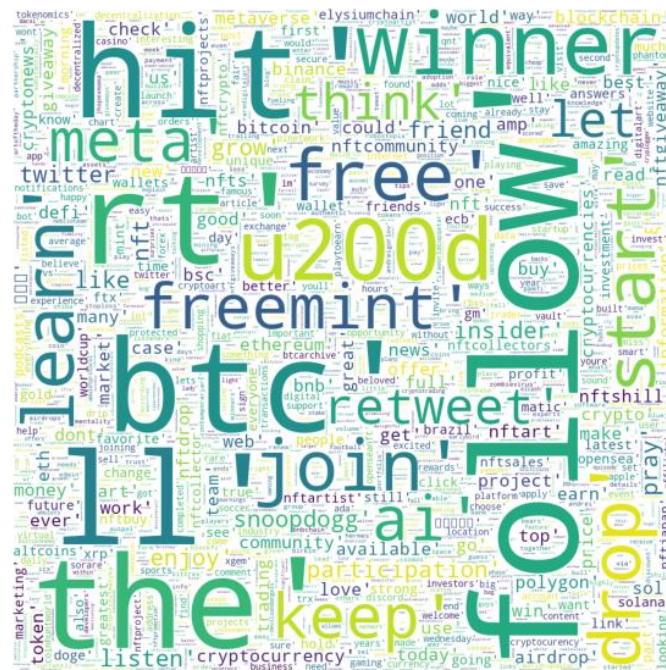
Lampiran 46. Sebaran Kata Sentimen Netral (65:35)



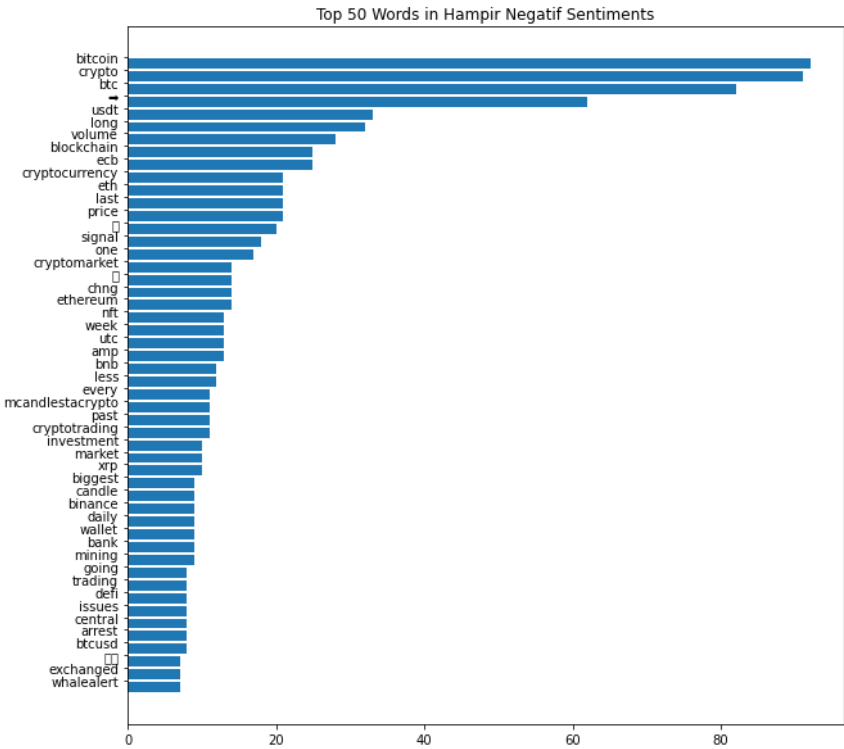
Lampiran 47. Sebaran Kata Wordcloud Sentimen Netral (65:35)



Lampiran 48. Sebaran Kata Sentimen Hampir Positif (65:35)



Lampiran 49. Sebaran Kata Sentimen Wordcloud Hampir Positif (65:35)



Lampiran 50. Sebaran Kata Sentimen Hampir Negatif (65:35)





```

komentar = ""
for itemset in frequent_itemsets:
    if set(itemset.items).issubset(komentar):
        sentimen = itemset.support
        break
    prediksi_hampirnegatif.append(sentimen)

# Menghitung akurasi data Latih netral
akurasi_latih_netral = np.mean(prediksi_netral == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_netral)

# Menghitung akurasi data Latih positif
akurasi_latih_positif = np.mean(prediksi_positif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_positif)

# Menghitung akurasi data Latih negatif
akurasi_latih_negatif = np.mean(prediksi_negatif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_negatif)

# Menghitung akurasi data Latih hampir positif
akurasi_latih_hampirpositif = np.mean(prediksi_hampirpositif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_hampirpositif)

# Menghitung akurasi data Latih hampir negatif
akurasi_latih_hampirnegatif = np.mean(prediksi_hampirnegatif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_hampirnegatif)

# Menghitung rata-rata total akurasi data Latih
rata_rata_akurasi_latih = np.mean(akurasi_latih)

# Menampilkan hasil
print("Rata-rata akurasi data latih: %.2f%%" % (rata_rata_akurasi_latih * 100))

```

Rata-rata akurasi data latih: 20.00%

## Lampiran 53. Akurasi Apriori Sentimen Latih (70:30)

```

komentar = ""
for itemset in frequent_itemsets:
    if set(itemset.items).issubset(komentar):
        sentimen = itemset.support
        break
    prediksi_hampirnegatif.append(sentimen)

# Menghitung akurasi data Latih netral
akurasi_latih_netral = np.mean(prediksi_netral == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_netral)

# Menghitung akurasi data Latih positif
akurasi_latih_positif = np.mean(prediksi_positif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_positif)

# Menghitung akurasi data Latih negatif
akurasi_latih_negatif = np.mean(prediksi_negatif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_negatif)

# Menghitung akurasi data Latih hampir positif
akurasi_latih_hampirpositif = np.mean(prediksi_hampirpositif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_hampirpositif)

# Menghitung akurasi data Latih hampir negatif
akurasi_latih_hampirnegatif = np.mean(prediksi_hampirnegatif == data_latih["sentimen"])
akurasi_latih.append(akurasi_latih_hampirnegatif)

# Menghitung rata-rata total akurasi data Latih
rata_rata_akurasi_latih = np.mean(akurasi_latih)

# Menampilkan hasil
print("Rata-rata akurasi data latih: %.2f%%" % (rata_rata_akurasi_latih * 100))

```

Rata-rata akurasi data latih: 20.00%

## Lampiran 54. Akurasi Apriori Sentimen Latih (65:35)

```

if set(itemset.items).issubset(komentar):
    sentimen = itemset.support
    break
    prediksi_hampirnegatif.append(sentimen)

# Menghitung akurasi data Latih netral
akurasi_uji_netral = np.mean(prediksi_netral == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_netral)

# Menghitung akurasi data Latih positif
akurasi_uji_positif = np.mean(prediksi_positif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_positif)

# Menghitung akurasi data Latih negatif
akurasi_uji_negatif = np.mean(prediksi_negatif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_negatif)

# Menghitung akurasi data Latih hampir positif
akurasi_uji_hampirpositif = np.mean(prediksi_hampirpositif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_hampirpositif)

# Menghitung akurasi data Latih hampir negatif
akurasi_uji_hampirnegatif = np.mean(prediksi_hampirnegatif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_hampirnegatif)

# Menghitung rata-rata total akurasi data Latih
rata_rata_akurasi_uji = np.mean(akurasi_uji)

# Menampilkan hasil
print("Rata-rata akurasi data uji: %.2f%%" % (rata_rata_akurasi_uji * 100))

```

Rata-rata akurasi data uji: 20.00%

## Lampiran 55. Akurasi Apriori Sentimen Uji (75:25)

```
akurasi_uji_netral = np.mean(prediksi_netral == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_netral)

# Menghitung akurasi data latih positif
akurasi_uji_positif = np.mean(prediksi_positif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_positif)

# Menghitung akurasi data latih negatif
akurasi_uji_negatif = np.mean(prediksi_negatif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_negatif)

# Menghitung akurasi data latih hampir positif
akurasi_uji_hampirpositif = np.mean(prediksi_hampirpositif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_hampirpositif)

# Menghitung akurasi data latih hampir negatif
akurasi_uji_hampirnegatif = np.mean(prediksi_hampirnegatif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_hampirnegatif)

# Menghitung rata-rata total akurasi data latih
rata_rata_akurasi_uji = np.mean(akurasi_uji)

# Menampilkan hasil
print("Rata-rata akurasi data uji: %.2f%%" % (rata_rata_akurasi_uji * 100))
```

Rata-rata akurasi data uji: 20.00%

## Lampiran 56. Akurasi Apriori Uji Sentimen (70:30)

```
sentimen = itemset.support
break
prediksi_hampirnegatif.append(sentimen)

# Menghitung akurasi data latih netral
akurasi_uji_netral = np.mean(prediksi_netral == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_netral)

# Menghitung akurasi data latih positif
akurasi_uji_positif = np.mean(prediksi_positif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_positif)

# Menghitung akurasi data latih negatif
akurasi_uji_negatif = np.mean(prediksi_negatif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_negatif)

# Menghitung akurasi data latih hampir positif
akurasi_uji_hampirpositif = np.mean(prediksi_hampirpositif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_hampirpositif)

# Menghitung akurasi data latih hampir negatif
akurasi_uji_hampirnegatif = np.mean(prediksi_hampirnegatif == data_uji["sentimen"])
akurasi_uji.append(akurasi_uji_hampirnegatif)

# Menghitung rata-rata total akurasi data latih
rata_rata_akurasi_uji = np.mean(akurasi_uji)

# Menampilkan hasil
print("Rata-rata akurasi data uji: %.2f%%" % (rata_rata_akurasi_uji * 100))
```

Rata-rata akurasi data uji: 20.00%

## Lampiran 57. Akurasi Apriori Uji Sentimen (65:35)

```
In [12]: import numpy as np
from sklearn.svm import SVC

#Lin_svm=svm.LinearSVC()
lin_svm = svm.SVC(kernel='linear',C=1)
lin_svm.fit(train_vectors, data_train['sentimen_emot'])
lin_svm.fit(test_vectors, data_test['sentimen_emot'])
y_train_hat=lin_svm.predict(train_vectors)
y_test_hat=lin_svm.predict(test_vectors)

print("train accuracy_analisa emoticon= %.0f%%" % (np.mean(y_train_hat == data_train['sentimen_emot']) * 100))
print("test accuracy_analisa emoticon= %.0f%%" % (np.mean(y_test_hat == data_test['sentimen_emot']) * 100))
```

train accuracy\_analisa emoticon= 71%  
test accuracy\_analisa emoticon= 92%

## Lampiran 58. Hasil Penghitungan Akurasi Emotikon Sentimen Klasifikasi SVM (75:35)

```
In [12]: import numpy as np
from sklearn.svm import SVC

#lin_svm=svm.LinearSVC()
lin_svm= svm.SVC(kernel='linear',C=1)
lin_svm.fit(train_vectors, data_train['sentimen_emot'])
lin_svm.fit(test_vectors, data_test['sentimen_emot'])
y_train_hat=lin_svm.predict(train_vectors)
y_test_hat=lin_svm.predict(test_vectors)

print("train accuracy_analisa emoticon= %.0f%%" % (np.mean(y_train_hat == data_train['sentimen_emot']) * 100))
print("test accuracy_analisa emoticon= %.0f%%" % (np.mean(y_test_hat == data_test['sentimen_emot']) * 100))
```

train accuracy\_analisa emoticon= 72%  
test accuracy\_analisa emoticon= 93%

### Lampiran 59. Hasil Penghitungan Akurasi Emotikon Sentimen Klasifikasi SVM (70:30)

```
In [9]: import numpy as np
from sklearn.svm import SVC

#lin_svm=svm.LinearSVC()
lin_svm= svm.SVC(kernel='linear',C=1)
lin_svm.fit(train_vectors, data_train['sentimen'])
lin_svm.fit(test_vectors, data_test['sentimen'])
y_train_hat=lin_svm.predict(train_vectors)
y_test_hat=lin_svm.predict(test_vectors)

print("train accuracy= %.0f%%" % (np.mean(y_train_hat == data_train['sentimen']) * 100))
print("test accuracy= %.0f%%" % (np.mean(y_test_hat == data_test['sentimen']) * 100))
```

train accuracy= 78%  
test accuracy= 95%

### Lampiran 60. Hasil Penghitungan Akurasi Emotikon Sentimen Klasifikasi SVM (65:35)