TUGAS AKHIR – KS184822

ANALISIS SENTIMEN NASABAH PADA LAYANAN PERBANKAN MENGGUNAKAN METODE REGRESI LOGISTIK BINER, *NAÏVE BAYES CLASSIFIER* (NBC), DAN *SUPPORT VECTOR MACHINE* (SVM)

ERNA DWI NURINDAH SARI
NRP 062117 4500 0034

Dosen Pembimbing
Irhamah, M.Si., Ph.D

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

TUGAS AKHIR – KS184822

ANALISIS SENTIMEN NASABAH PADA LAYANAN PERBANKAN MENGGUNAKAN METODE REGRESI LOGISTIK BINER, *NAÏVE BAYES CLASSIFIER* (NBC), DAN *SUPPORT VECTOR MACHINE* (SVM)

ERNA DWI NURINDAH SARI
NRP 062117 4500 0034

Dosen Pembimbing
Irhamah, M.Si., Ph.D

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

FINAL PROJECT – KS184822

CUSTOMER SENTIMEN ANALYSIS OF BANK SERVICE USING BINARY LOGISTIC REGRESSION, NAÏVE BAYES CLASSIFIER (NBC), AND SUPPORT VECTOR MACHINE (SVM)

ERNA DWI NURINDAH SARI
SN 062117 4500 0034

Supervisor
Irhamah, M.Si., Ph.D

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

LEMBAR PENGESAHAN

ANALISIS SENTIMEN NASABAH PADA LAYANAN PERBANKAN DENGAN MENGGUNAKAN METODE REGRESI LOGISTIK BINER, *NAÏVE BAYES CLASSIFIER* (NBC), DAN *SUPPORT VECTOR MACHINE* (SVM)

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Statistika
pada
Program Studi Sarjana Departemen Statistika
Fakultas Metematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Oleh :
Erna Dwi Nurindah Sari
NRP. 062117 4500 0034

Disetujui oleh Pembimbing:
Irhamah, M.Si., Ph.D
NIP. 19780406 200112 2 002

( )

Mengetahui,
Kepala Departemen Statistika

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2019

Halaman ini sengaja dikosongkan

ANALISIS SENTIMEN NASABAH PADA LAYANAN PERBANKAN MENGGUNAKAN METODE REGRESI LOGISTIK BINER, *NAÏVE BAYES CLASSIFIER* (NBC), DAN *SUPPORT VECTOR MACHINE* (SVM)

Nama : **Erna Dwi Nurindah Sari**
NRP : **062117 4500 0034**
Departemen : **Statistika**
Dosen Pembimbing : **Irhamah, M.Si., Ph.D**

Abstrak

Di era globalisasi, semua aktivitas yang dilakukan tidak dapat terlepas dari teknologi. Dengan menggunakan Twitter pendapat masyarakat mengenai layanan dari perbankan dapat diketahui. Pada penelitian ini pendapat nasabah akan dibedakan menjadi sentimen positif atau negatif sehingga hasil analisa sentimen tersebut dapat dijadikan evaluasi sebagai peningkatan layanan pada para nasabahnya. Klasifikasi sentimen positif dan sentimen negatif dilakukan dengan beberapa metode yakni menggunakan Regresi Logistik Biner yang merupakan salah satu metode konvensional, metode Naïve Bayes Classifier yakni metode yang sederhana namun memiliki ketepatan klasifikasi yang baik, dan metode Support Vector Machine (SVM) yang dapat mengklasifikasikan sentimen dengan beberapa jenis kernel. Data yang digunakan pada penelitian ini diklasfikasikan secara manual dan dengan menggunakan kamus lexicon. Karena jumlah data sentiment yang tidak seimbang maka dilakukan SMOTE pada data klasifikasi manual. Hasil dari penelitian menunjukkan bahwa metode terbaik untuk mengklasifikasikan sentimen pada layanan BRI adalah SMOTE-SVM kernel RBF, sedangkan untuk Bank Mandiri adalah SMOTE-NBC karena memiliki nilai AUC paling tinggi.

Kata Kunci : ***Naïve Bayes Classifier, Perbankan, Regresi Logistik Biner, Sentimen Negatif, Sentimen Positif, SVM.***

Halaman ini sengaja dikosongkan

CUSTOMER SENTIMEN ANALYSIS OF BANK SERVICE USING BINARY LOGISTIC REGRESSION, NAÏVE BAYES CLASSIFIER (NBC), AND SUPPORT VECTOR MACHINE (SVM)

Name : **Erna Dwi Nurindah Sari**
Student Number : **062117 4500 0034**
Department : **Statistics**
Supervisor : **Irhamah, M.Si., Ph.D**

Abstract

In this era of globalization, all activities cannot be separated from technology. At this time, public opinion regarding services from banks can be known by using Twitter. In this study, the customer's opinion will be divided into positive and negative sentiments so that the results of the sentiment analysis can be used as an evaluation to its customers. The classification of positive and negative sentiment is done by several methods namely using Binary Logistic Regression is one of conventional method, Naïve Bayes Classifier method which is a simple method but has good classification accuracy, and Support Vector Machine (SVM) method can classifiy sentiments with several types of kernels. In this study, customer sentiment are classified with manual process and used lexicon based. Because of the imbalancing data, then the manual classification will duplicated using SMOTE method. The result of the study shows that the best method for sentiment classification of BRI service is SMOTE-SVM with RBF kernel, while for Mandiri Bank is SMOTE-NBC because on that method give the highest result AUC.

Keywords : ***Naïve Bayes Classifier, Banking, Binary Logistic Regression, Negative Sentiment, Positive Sentiment, SVM.***

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul "Analisis Sentimen Nasabah Pada Layanan Perbankan dengan Menggunakan Metode Regresi Logistik Biner, *Naïve Bayes Classifier* (NBC), dan *Support Vector Machine* (SVM)" untuk memenuhi persyaratan akademis Departemen Statistika FMKSD Institut Teknologi Sepuluh Nopember.

Dalam penyusunan tugas akhir ini penulis menyampaikan ucapan terima kasih kepada pihak – pihak yang membantu dalam penulisan laporan ini, yaitu:

1. Irhamah, M.Si., Ph.D., selaku dosen pembimbing tugas akhir yang tidak ada lelahnya dalam memberikan pengarahan dan pembimbingannya selama menyelesaikan masalah dalam penyusunan tugas akhir ini.
2. Orang tua yang sudah membantu memberikan semangat dan doanya agar tugas akhir ini cepat selesai.
3. Dr. Suhartono sebagai Kepala Departemen Statistika yang telah memberikan fasilitas untuk penyelesaian tugas akhir ini.
4. Dr. Kartika Fithriasari, M.Si dan Pratnya P. Oktaviana, M.Si selaku dosen penguji yang sudah banyak memberikan saran dan kritik dalam penulisan buku tugas akhir ini.
5. M. Sjahid Akbar, S.Si., M.Si. sebagai dosen wali yang telah memberikan dukungan dan motivasi kepada penulis.
6. Kakak dan saudara kembar yang sudah mendukung dalam hal apapun.
7. Teman-teman sebimbingan yang sudah membantu dalam menyelesaikan masalah.
8. Vriesia, Devi, Ninda, Lely, Vida, Razty yang selalu memberikan dukungan dan motivasi untuk semua dalam berjuang menyelesaikan tugas akhir.

9. Para teman-teman ”Lintas Jalur*”* yang selalu mengingatkan untuk segera menyelesaikan tugas akhir ini dengan baik.
10. Para dosen dan staf yang ada di Departmen Statistika.
11. Semua teman-teman yang tidak dapat kami sebutkan satu persatu yang telah memberikan saran atas penulisan tugas akhir ini.

Pada akhirnya penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan kesalahan. Untuk itu penulis mengharapkan kritik dan saran dari para pembaca. Semoga laporan ini dapat bermanfaat bagi kita semua.

Surabaya, 15 Juli 2019

Penulis

DAFTAR ISI

Halaman

ABSTRAK ..... **v**
ABSTRACT ..... **vii**
DAFTAR ISI ..... **xi**
DAFTAR TABEL ..... **xiii**
DAFTAR GAMBAR ..... **xvii**
DAFTAR LAMPIRAN ..... **xix**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Perumusan Masalah 5
1.3 Tujuan 5
1.4 Manfaat 5
1.5 Batasan Masalah 5
BAB II TINJAUAN PUSTAKA **7**
2.1 *Text Mining* 7
2.2 *Text Preprocessing* 7
2.3 TF-IDF 8
2.4 Regresi Logistik Biner 9
2.4.1 Estimasi Parameter 10
2.4.2 Uji Serentak 12
2.4.3 Uji Individu 13
2.5 *Naïve Bayes Classifier* 13
2.6 *Support Vector Machine* 15
2.6.1 *Support Vector Machine Linear* 16
2.6.2 *Support Vector Machine Non Linear* 17
2.6.3 Pemilihan Parameter Fungsi Kernel SVM 17
2.7 *K-fold Cross Validation* 19
2.8 SMOTE 19
2.9 Ketepatan Klasifikasi 22
2.10 *Word Cloud* 23
2.11 Bank Rakyat Indonesia (BRI) 24
2.12 Bank Mandiri 24
BAB III METODOLOGI PENELITIAN **27**
3.1 Sumber Data 27

3.2 Variabel Penelitian dan Struktur Data 27
3.3 Langkah Analisis 28
3.4 Diagram Alir 31
BAB IV ANALISIS DAN PEMBAHASAN 35
4.1 Praproses Teks 35
4.2 Karakteristik *Tweet* 36
4.3 Klasifikasi pada data BRI 38
4.3.1 TF-IDF untuk data BRI 39
4.3.2 Klasifikasi data BRI dengan Regresi Logistik Biner 40
4.3.3 Klasifikasi data BRI dengan *Naïve Bayes Classifier* 47
4.3.4 Klasifikasi data BRI dengan *Support Vector Machine* Kernel Linier 52
4.3.5 Klasifikasi data BRI *Support Vector Machine* Kernel RBF 56
4.4 Klasifikasi pada data Bank Mandiri 60
4.4.1 TF-IDF untuk data Bank Mandiri 61
4.4.2 Klasifikasi untuk Bank Mandiri dengan Regresi Logistik Biner 62
4.4.3 Klasifikasi untuk Bank Mandiri dengan *Naïve Bayes Classifier* 68
4.4.4 Klasifikasi data Bank Mandiri dengan *Support Vector Machine* Kernel Linier 72
4.4.5 Klasifikasi data Bank Mandiri dengan *Support Vector Machine* Kernel RBF 76
4.5 Pemilihan Metode Terbaik 80
4.6 Visualisasi *Word Cloud* Pada Setiap Bank 83
4.6.1 *Word Cloud* BRI 83
4.6.2 *Word Cloud* Bank Mandiri 85
BAB V KESIMPULAN DAN SARAN 89
5.1 Kesimpulan 89
5.2 Saran 89
DAFTAR PUSTAKA 91
LAMPIRAN 94

DAFTAR TABEL

Halaman

Tabel 2.1. Sampel untuk SMOTE 20
Tabel 2.2. *Confusion Matrix* 22
Tabel 2.3. Kebaikan Nilai AUC 23
Tabel 3.1. Variabel Penelitian 27
Tabel 3.2. Struktur Data 28
Tabel 4.1. Contoh Praproses Teks 35
Tabel 4.5. Frekuensi Kemunculan Kata pada Tweet BRI 40
Tabel 4.9. Performa Klasifikasi Regresi Logistik Biner BRI pada Data Awal 44
Tabel 4.10. *Confusion Matrix* untuk Regresi Logistik Biner BRI pada Data *Testing* Awal 44
Tabel 4.11. Performa Klasifikasi Regresi Logistik Biner BRI pada Data SMOTE 45
Tabel 4.12. *Confusion Matrix* untuk Regresi Logistik Biner BRI pada Data *Testing* SMOTE 45
Tabel 4.13. Performa Klasifikasi Regresi Logistik Biner BRI pada Data *Lexicon* 46
Tabel 4.14. *Confusion Matrix* untuk Regresi Logistik Biner BRI pada Data Testing *Lexicon* 46
Tabel 4.15. Performa Klasifikasi NBC BRI pada Data Awal 49
Tabel 4.16. *Confusion Matrix* untuk NBC BRI pada Data *Testing* Awal 49
Tabel 4.17. Performa Klasifikasi NBC BRI dengan Menggunakan Data SMOTE 50
Tabel 4.18. *Confusion Matrix* untuk NBC BRI pada Data *Testing* SMOTE 50
Tabel 4.19. Performa Klasifikasi NBC BRI pada Data *Lexicon* 51
Tabel 4.20. *Confusion Matrix* untuk NBC BRI pada Data *Testing Lexicon* 51
Tabel 4.21. Performa Klasifikasi SVM Linier BRI pada Data Awal 53
Tabel 4.22. *Confusion Matrix* untuk SVM Linier BRI pada Data *Testing* Awal 53

Tabel 4.23. Performa Klasifikasi SVM Linier BRI pada Data SMOTE.....54
Tabel 4.24. *Confusion Matrix* untuk SVM Kernel Linear BRI Data *Testing* SMOTE 54
Tabel 4.25. Performa Klasifikasi SVM Linier BRI pada Data Lexicon 55
Tabel 4.26. *Confusion Matrix* untuk SVM Kernel Linear BRI Data *Testing* Lexicon.....55
Tabel 4.27. Performa Klasifikasi SVM RBF BRI dengan Menggunakan Data Awal 57
Tabel 4.28. *Confusion Matrix* untuk SVM RBF BRI pada Data *Testing* Awal.....57
Tabel 4.29. Performa Klasifikasi SVM RBF BRI pada Data SMOTE.....58
Tabel 4.30. *Confusion Matrix* untuk SVM RBF BRI pada Data *Testing* SMOTE.....59
Tabel 4.31. Performa Klasifikasi SVM RBF BRI pada Data *Lexicon* 59
Tabel 4.32. *Confusion Matrix* untuk SVM RBF BRI pada Data *Testing Lexicon*.....60
Tabel 4.33. TF-IDF Bank Mandiri.....61
Tabel 4.34. Frekuensi Kemunculan Kata pada Tweet Mandiri..61
Tabel 4.37. Uji Signifikansi Parameter Bank Mandiri.....64
Tabel 4.38. Performa Klasifikasi Regresi Logistik Biner Bank Mandiri pada Data Awal.....65
Tabel 4.39. *Confusion Matrix* untuk Regresi Logistik Biner Bank Mandiri pada Data *Testing* Awal.....66
Tabel 4.40. Performa Klasifikasi Regresi Logistik Biner Bank Mandiri pada Data SMOTE.....66
Tabel 4.41. *Confusion Matrix* untuk Regresi Logistik Biner Bank Mandiri pada Data *Testing* SMOTE.....67
Tabel 4.42. Performa Klasifikasi Regresi Logistik Biner Bank Mandiri pada Data *Lexicon*.....67
Tabel 4.43. *Confusion Matrix* untuk Regresi Logistik Biner Bank Mandiri pada Data *Testing Lexicon*.....68

Tabel 4.44. Performa Klasifikasi NBC Bank Mandiri pada Data Awal 69
Tabel 4.45. *Confusion Matrix* untuk NBC Bank Mandiri pada Data *Testing* Awal 70
Tabel 4.46. Performa Klasifikasi NBC Bank Mandiri pada Data SMOTE 70
Tabel 4.47. *Confusion Matrix* untuk NBC Bank Mandiri pada Data *Testing* SMOTE 71
Tabel 4.48. Performa Klasifikasi NBC Bank Mandiri pada Data *Lexicon* 71
Tabel 4.49. *Confusion Matrix* untuk NBC Bank Mandiri pada Data *Testing Lexicon* 72
Tabel 4.50. Performa Klasifikasi SVM Linier Bank Mandiri pada Data Awal 72
Tabel 4.51. *Confusion Matrix* untuk SVM Linier Bank Mandiri pada Data *Testing* Awal 73
Tabel 4.52. Performa Klasifikasi SVM Linier Bank Mandiri pada Data SMOTE 73
Tabel 4.53. *Confusion Matrix* untuk SVM Kernel Linear Bank Mandiri Data *Testing* SMOTE 74
Tabel 4.54. Performa Klasifikasi SVM Linier Bank Mandiri pada Data *Lexicon* 74
Tabel 4.55. *Confusion Matrix* untuk SVM Kernel Linear Bank Mandiri Data *Testing Lexicon* 75
Tabel 4.56. Performa Klasifikasi SVM RBF Bank Mandiri pada Data Awal 76
Tabel 4.57. *Confusion Matrix* untuk SVM RBF Bank Mandiri pada Data *Testing* Awal 77
Tabel 4.58. Performa Klasifikasi SVM RBF Bank Mandiri pada Data SMOTE 77
Tabel 4.59. *Confusion Matrix* untuk SVM RBF Bank Mandiri pada Data Testing SMOTE 78
Tabel 4.60. Performa Klasifikasi SVM RBF Bank Mandiri pada Data *Lexicon* 78

Tabel 4.61. *Confusion Matrix*untuk SVM RBF Bank Mandiri pada Data *Testing Lexicon*.....79
Tabel 4.62. Pemilihan Metode Terbaik Pada Pelayanan BRI....80
Tabel 4.63. Pemilihan Metode Terbaik Pada Pelayanan Bank Mandiri 81

DAFTAR GAMBAR

Halaman

Gambar 3.1 Diagram Alir Penelitian. 31
Gambar 3.2 Diagram Alir Klasifikasi NBC.....34
Gambar 4.1 Frekuensi Data *Tweet* Setiap Sentimen Untuk Data dengan Klasifikasi Manual.....37
Gambar 4.2 Frekuensi Data *Tweet* Setiap Sentimen Untuk Data dengan Klasifikasi *Lexicon* 37
Gambar 4.2 Word Cloud Unigram BRI.....83
Gambar 4.3 *Word Cloud* Bigram Sentimen Negatif BRI.....84
Gambar 4.4 *Word Cloud* Unigram Bank Mandiri.....85
Gambar 4.5 *Word Cloud* Bigram Sentimen Negatif Mandiri ..86

Halaman ini sengaja dikosongkan

DAFTAR LAMPIRAN

Halaman

Lampiran 1. *Syntax Crawling* 95
Lampiran 2. *Syntax* Praproses Data 96
Lampiran 3. *Syntax* Klasifikasi 100
Lampiran 4. *Syntax* Klasifikasi Regresi Logistik Biner 101
Lampiran 5. *Syntax* Klasifikasi *Naïve Bayes Classifier* 102
Lampiran 6. *Syntax* Klasifikasi SVM 103
Lampiran 7. *Syntax* SMOTE 105
Lampiran 8. *Syntax Wordcloud* 107
Lampiran 9. *Syntax* Estimasi Parameter untuk Regresi Logistik Biner 108
Lampiran 10. Hasil Tuning Parameter 109
Lampiran 11. Surat Pernyataan 111

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada saat ini perkembangan teknologi sudah sangat berkembang pesat. Di era globalisasi ini, semua aktivitas yang dilakukan tidak dapat terlepas dari teknologi. Digitalisasi dalam banyak aspek membuat sumber daya manusia terbantu dan lebih mudah dalam menangani berbagai macam pekerjaan sehingga kualitas layanan yang dihasilkan akan meningkat, salah satunya adalah bank. Bank merupakan lembaga yang bergerak dan mengelola keuangan yang berasal dari para nasabahnya. Nasabah merupakan seseorang yang menggunakan jasa dari perbankan. Kepuasan pelayanan nasabah menjadi hal yang penting untuk diperhatikan pada industri perbankan. Oleh karena itu, pelayanan yang baik dapat memberikan kepuasan nasabah dalam bertransaksi di industri perbankan. Ketepatan dan kecepatan layanan yang disuguhkan oleh perbankan akan menjadi hal yang sangat diperlukan pada era sekarang ini.

Di Indonesia terdapat beberapa perusahaan perbankan antara lain adalah Bank Rakyat Indonesia (BRI), Bank Mandiri, Bank Central Asia (BCA), dan lain-lain. Bank dalam melakukan kegiatan usaha dan memperluas jaringan kantornya harus sesuai dengan kapasitas dasar yang dimiliki bank, yaitu modal inti. Dengan beroperasi sesuai dengan kapasitasnya, bank dipercaya dapat memiliki ketahanan yang lebih baik dan akan lebih efisien karena kegiatannya terfokus pada produk dan aktivitas yang memang menjadi keunggulannya. Berdasarkan modal intinya kegiatan usaha bank dikelompokkan menjadi empat yaitu BUKU 1, BUKU 2, BUKU 3, dan BUKU 4. Sejalan dengan besaran modal intinya, kegiatan usaha yang terdapat pada BUKU 1 lebih bersifat layanan dasar perbankan (*basic banking services*). Kegiatan usaha pada BUKU 2 lebih luas daripada BUKU 1 dan demikian seterusnya hingga BUKU 4 yang mencakup kegiatan usaha penuh dan kompleks (OJK, 2018). Bank yang masuk dalam kategori

BUKU 4 di Indonesia adalah BRI, Bank Mandiri, BCA, BNI, dan CIMB Niaga. Pemilihan bank pada penelitian ini didasarkan pada keuntungan yang dimiliki dan jumlah nasabah dari masing-masing bank. BRI merupakan bank yang memiliki jumlah pelanggan paling banyak di Indonesia dengan jumlah nasabah mencapai sekitar 50.000.000 nasabah. Menurut Forbes (2018) bank yang memiliki keuntungan paling tinggi yakni adalah BRI dan selanjutnya Bank Mandiri untuk kategori bank yang dikelola oleh pemerintahan. Jika dilihat dari harga saham, pada harga penutupan 11 Februari 2018 harga saham tertinggi untuk bank umum milik pemerintah terdapat pada Bank Mandiri dengan harga saham sebesar 7.625,00 (Financial, 2018). BRI merupakan bank dengan total penjualan mencapai $9,3 B sedangkan Bank Mandiri sendiri memiliki total penjualan sebesar $ 8,9 B. Berdasarkan hal tersebut BRI dan Bank Mandiri adalah bank yang dipilih dalam melakukan penelitian ini.

Berbagai macam kondisi layanan perbankan tersebut akan menimbulkan berbagai respon dari nasabah. Jika terjadi kesalahan pada dalam melakukan transaksi di perbankan nasabah pada saat ini akan cenderung melakukan protes atau mengajukan keluhan ke pihak yang bersangkutan. Namun jika pelayanan yang dilakukan oleh perbankan cepat dan memuaskan maka masyarakat akan cenderung melakukan apresiasi. Pada saat ini jejaring sosial merupakan sarana yang banyak dipilih masyarakat dalam menceritakan berbagai pendapat dan keluh kesahnya. Salah satu jejaring sosial yang banyak digunakan adalah Twitter. Twitter merupakan layanan jejaring sosial dan mikroblog daring yang memungkinkan penggunanya untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter akan tetapi telah bertambah hingga 280 karakter yang dikenal dengan sebutan kicauan (*tweet*). Oleh karena itu, dengan menggunakan Twitter dapat diketahui pendapat nasabah mengenai layanan perbankan. Pendapat masyarakat tersebut nantinya akan dibedakan menjadi sentimen yang bersifat positif atau negatif sehingga berdasarkan sentimen tersebut dapat dijadikan evaluasi dan selanjutnya perusahaan

perbankan dapat melakukan peningkatan ataupun perbaikan dalam memberikan layanan kepada nasabahnya.

Klasifikasi sentimen positif dan sentimen negatif dilakukan dengan metode statistika yakni menggunakan *text mining. Text mining* merupakan suatu cabang ilmu dari statistika yang mengolah teks menjadi suatu data. Proses *text mining* ini lebih banyak dilakukan pada suatu pengelompokkan data. Salah satu cara untuk pengelompokkan data adalah dengan menggunakan Regresi Logistik Biner. Metode Regresi Logistik Biner merupakan metode klasik yang digunakan untuk mengetahui pola hubungan antara variabel respon yang bersifat biner yakni terdiri dari 0 dan 1 dengan variabel prediktornya (Hosmer & Stanley, 2000). Metode lain yang digunakan adalah *Naïve Bayes Classifier* dan *Support Vector Machine* (SVM). *Naïve Bayes Classifier* adalah suatu klasifikasi yang berdasarkan pada teorema Bayes yang bertujuan dalam menghitung peluang pada tiap kelas serta memiliki asumsi bahwa hubungan antar kelas adalah independen (Gorunescu, 2011) sedangkan *Support Vector Machine* (SVM) memiliki ide dasar bahwa menemukan fungsi pemisah (*hyperlane*) yang nantinya dapat memisahkan dua kelas secara optimal (Tan, Steinbach, Karpatne, & Kumar, 2018).

Penelitian sebelumnya pernah dilakukan oleh Anwar (2015) mengenai analisis sentimen pada akun Twitter provider telekomunikasi menggunakan metode klasifikasi *Support Vector Machine* (*SVM*) dan metode klasifikasi *Naïve Bayes*, hasil yang diperoleh yaitu akurasi sebesar 96% dari 300 data yang diuji menggunakann metode pengujian *confusion matrix*. Selanjutnya terdapat penelitian Manalu (2014) tentang analisis sentimen pada Twitter menggunakan *text mining* dengan metode yang digunakan adalah *Naïve Bayes Classifier,* hasil penelitiannya menghasilkan sebuah sistem yang dapat mengklasifikasikan sentimen secara otomatis dengan hasil akurasi 100 *tweet* mecapai 93% dengan 2700 data *training.* Peneliti lain yang pernah melakukan penelitian mengenai analisis sentimen adalah Karimah (2018) mengenai analisis kompetitif sosial media menggunakan metode klasifikasi

Naïve Bayes Classifier (NBC) dan *Support Vector Machine* (SVM) dengan studi kasus di industri transportasi umum taksi, hasil yang diperoleh adalah metode yang paling bagus digunakan adalah *Support Vector Machine Kernel RBF* dengan rata-rata akurasi 81,7% untuk BlueBird Taxi, 77,6% untuk GoCar, dan 78,5% untuk GrabCar. Selanjutnya terdapat Mayasari (2018) yang melakukan penelitian mengenai *text mining* pada akun resmi Pemerintah Kota Surabaya dengan metode Regresi Logistik Biner, *Support Vector Machine* (*SVM*), dan *Naïve Bayes Classifier* (NBC) dan memperoleh hasil bahwa metode SVM denga kernel RBF memiliki hasil analisis terbaik dibandingkan dengan NBC dan Regresi Logistik Biner. Peneliti lain yang pernah melakukan penelitian serupa yakni Kurniawan (2017) yakni tentang implementasi *text mining* pada analisis sentimen pengguna Twitter terhadap media *mainstream* menggunakan *Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) dan diperoleh hasil bahwa performa metode SVM lebih baik dalam mengklasifikasikan data daripada metode NBC serta kernel RBF memiliki ketepatan klasifikasi lebih baik daripada kernel *linear*.

Pada penelitian ini, kepuasan nasabah merupakan hal yang sangat diperhatikan sehingga ketepatan dan kecepatan layanan pada perbankan akan menimbulkan berbagai macam respon dari nasabah. Respon tersebut akan bersifat positif dan negatif sesuai dengan pelayanan dari perbankan. Terdapat dua variabel yang digunakan pada penelitian ini, yakni variabel dependen dan independen. Variabel dependen yang digunakan adalah klasifikasi dari hasil sentimen *tweet* yang terbagi menjadi sentimen positif dan sentimen negatif sedangkan variabel independennya adalah kata kunci berdasarkan *tweet* setelah melalui *text preprocessing*. Metode yang digunakan untuk mengklasifikasikan *tweet* menjadi sentimen positif dan sentimen negatif adalah dengan menggunakan Regresi Logistik Biner, *Naive Bayes Classifier*, dan *Support Vector Machine* sehingga tujuan dilakukannya penelitian ini adalah untuk mengklasifikasikan pendapat para nasabah mengenai layanan yang

diberikan oleh bank sehingga bank dapat melakukan evaluasi dalam hal pelayanan yang dimiliki.

1.2 Perumusan Masalah

Pada saat ini kepuasan nasabah merupakan hal yang sangat diperhatikan oleh perbankan sehingga ketepatan dan kecepatan layanan pada perbankan akan menimbulkan berbagai macam respon dari nasabah. Respon tersebut akan diklasifikasikan menjadi sentimen positif dan negatif sesuai dengan pelayanan dari perbankan. Pada saat ini banyak nasabah yang memilih untuk mengungkapkan responnya pada sosial media Twitter sehingga akan lebih mudah terpantau pendapat nasabah yang perlu ditindak lanjuti dengan cepat. Oleh karena itu, berdasarkan penjelasan tersebut, permasalahan yang akan dibahas pada penelitian ini adalah bagaimana klasifikasi sentimen nasabah menggunakan metode Regresi Logistik Biner, metode *Naive Bayes Classifier*, dan metode *Support Vector Machine*.

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk mengetahui hasil klasifikasi sentimen nasabah terhadap layanan perbankan dengan menggunakan metode Regresi Logistik Biner, metode *Naive Bayes Classifier*, dan metode *Support Vector Machine*.

1.4 Manfaat

Manfaat yang ingin didapatkan dari hasil penelitian ini adalah memberikan evaluasi terhadap perusahaan perbankan terhadap layanan yang disediakan oleh perbankan sehingga kedepannya perusahaan perbankan dapat melakukan peningkatan kualitas dalam hal pelayanan.

1.5 Batasan Masalah

Pada penelitian ini batasan masalah yang digunakan adalah akun Twitter @kontakBRI dan akun Twitter @mandiricare dengan

rentang waktu pengambilan *tweet* pada 7 Februari 2019 hingga 8 April 2019. *Tweet* yang diambil merupakan *tweet* berbahasa indonesia yang diambil menggunakan Twitter API. *Labeling* pada klasifikasi sentimen dilakukan menggunakan *lexicon based*. Pada metode *Support Vector Machine* jenis kernel yang digunakan adalah kernel *linear* dan kernel *RBF*.

BAB II
TINJAUAN PUSTAKA

Pada bab berikut, berisi mengenai tinjauan pustaka yang digunakan sebagai landasan yang digunakan dalam melakukan penelitian.

2.1 *Text Mining*

Text mining adalah suatu proses untuk mengekstraksi suatu pola untuk dapat dieksplorasi yang datanya berasal dari suatu teks. *Text mining* adalah suatu disiplin ilmu yang berdasarkan pada *information retrieval*, *data mining*, *machine learning*, ilmu statistika, dan lingusitik komputasi (Talib, Hanif, Ayesha, & Fatima, 2016). Tujuan dari *text mining* adalah menemukan informasi yang belum diketahui. Pada *text mining* pola diekstraksi dari Bahasa yang umum dan tidak terstruktur. Pada analisisnya *text mining* melibatkan pengenalan pola, penandaan, ektraksi informasi, teknik *data mining* yang terdiri dari *link*, analisis asosiasi, visualisasi, dan analisis prediksi. Tujuan menyeluruh dalam *text mining*, pada dasarnya adalah untuk mengubah teks menjadi data untuk dianalisis melalui aplikasi dengan suatu metode analisis (Kumar, 2015).

2.2 *Text Preprocessing*

Text Preprocessing adalah suatu tahap pertama dalam mengolah teks untuk digunakan dalam pengubahan dokumen menjadi data yang terstruktur sesuai dengan kebutuhannya agar dapat diolah lebih lanjut dalam proses *text mining* (Karimah, 2018). Tahapan dalam praproses teks adalah sebagai berikut.

a. *Cleansing*, merupakan proses membersihkan *tweet* dari kata yang tidak diperlukan untuk mengurangi *noise*.
b. *Case Folding*, merupakan proses untuk mengubah semua karakter teks menjadi huruf kecil serta menghilangkan tanda baca dan angka.

c. *Stemming*, merupakan proses untuk mendapatkan kata dasar dengan cara menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran).
d. *Stopwords*, merupakan kosakata yang bukan termasuk kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat.
e. *Tokenizing*, merupakan proses memecah yang semula berupa kalimat menjadi kata-kata.

2.3 TF-IDF

TF-IDF adalah suatu metode pembobotan yang banyak digunakan dalam membangun model vektor. Dengan menggunakan TF-IDF akan diketahui seberapa penting suatu kata terhadap kumpulan *tweet* berdasarkan frekuensi kemunculannya. TF-IDF bekerja dengan melibatkan perkalian antara *Term Frequency* (TF) dengan *Inverse Document Frequency* (IDF). TF memiliki tujuan untuk menunjukkan jumlah kemunculan sebuah kata pada suatu *tweet*. IDF memiliki tujuan untuk menghitung frekuensi kemunculan suatu kata pada seluruh *tweet* (Brownlee, 2019). Semakin tinggi nilai dari TF-IDF makan semakin jaran kemunculan suatu kata tersebut dalam sekumpulan *tweet*. Rumus yang digunakan dalam memperoleh nilai TF-IDF dituliskan dengan persamaan (2.1) dan persamaan (2.2)

$$w_{ij} = tf_{ij} \times idf \,, \tag{2.1}$$

$$idf = \log\left(\frac{N}{df_i}\right), \tag{2.2}$$

dimana w_{ij} merupakan bobot dari kata *i* pada *tweet* ke-*j*, *N* merupakan jumlah seluruh *tweet*, tf_{ij} merupakan jumlah munculnya kata *i* pada *tweet* ke-*j*, dan df_i adalah banyaknya *tweet* yang mengandung kata *i*.

2.4 Regresi Logistik Biner

Regresi logistik biner merupakan suatu metode analisis data yang digunakan untuk mencari hubungan antara variabel respon (*y*) yang bersifat biner dengan variabel (*x*) yang bersifat polikotomus (Hosmer & Stanley, 2000). Hasil dari variabel respon yang terdiri dari 2 kategori yaitu sukses dan gagal yang dinotasikan dengan $y=1$ untuk sukses dan $y=0$ untuk gagal. Oleh karena itu, variabel *y* mengikuti distribusi *Bernoulli* untuk setiap observasi tunggal. Fungsi probabilitas untuk setiap observasi dapat dituliskan pada persamaan (2.3)

$$f(y)=\pi^{y}(1-\pi)^{1-y};\ y=0,1, \tag{2.3}$$

dimana *y* adalah variabel respon, jika $y=0$ maka $f(y)=1-\pi$ dan jika $y=1$ maka $f(y)=\pi$. Fungsi regresi logistik dapat dituliskan pada persamaan (2.4)

$$f(z)=\frac{1}{1+e^{-z}} \text{ ekuivalen } f(y)=\frac{e^{z}}{1+e^{z}}, \tag{2.4}$$

dimana, $z=\beta_0+\beta_1 x_1+...+\beta_p x_p$ dengan *p* adalah banyaknya variabel prediktor. Nilai *z* antara $-\infty$ dan $+\infty$ sehingga nilai $f(z)$ terletak antara 0 dan 1 untuk setiap nilai *z* yang diberikan. Hal tersebut menunjukkan bahwa model logistik menggambarkan probabilitas atau risiko dari suatu objek sehingga model regresi dapat dituliskan pada persamaan (2.5)

$$\pi(x)=\frac{e^{(\beta_0+\beta_1 x_1)}}{1+e^{(\beta_0+\beta_1 x_1)}}. \tag{2.5}$$

β_p merupakan parameter ke-*p* yang diestmiasi. Untuk mempermudah pendugaan parameter regresi maka model regresi logistik pada persamaan (2.5) dapat diurutkan dengan menggunakan transformasi logit $\pi(x)$ sehingga diperoleh persamaan (2.6)

$$g(x)=\ln\left(\frac{\pi(x)}{1-\pi(x)}\right)=\beta_0+\beta_1 x_1. \tag{2.6}$$

Model tersebut merupakan fungsi linier dari parameter-parameternya. Pada regresi logistik, variabel respon diekspresikan sebagai $y = \pi(x) + \varepsilon$ dimana ε mempunyai salah satu dari kemungkinan dua nilai yaitu $y = \pi(x) + \varepsilon$ dengan peluang $\pi(x)$ jika $y = 1$ dan $\varepsilon = -\pi(x)$ dengan peluang $1 - \pi(x)$ jika dan mengikuti distribusi binomial dengan rataan nol dan varians $(\pi(x))(1 - \pi(x))$.

2.4.1 Uji Independensi

Pengujian independensi dapat digunakan untuk mengetahui ada tidaknya hubungan antara dua variable yang mana skala data dari kedua variable yaitu interval atau rasio. Hipotesis untuk pengujian independensi adalah sebagai berikut.

H_0 : tidak terdapat hubungan antara dua variabel

H_1 : terdapat hubungan antara dua variabel

Untuk statistik uji dalam independensi dapat menggunakan statistik uji *pearson correlation* yang ditunjukkan pada persamaan (2.7) jika data dari kedua variabel merupakan sama-sama rasio.

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}}, \qquad (2.7)$$

dimana r merupakan nilai *pearson correlation*, x_i merupakan variabel pertama ke-i, y_i merupakan variabel kedua ke-i, dan n merupakan banyaknya observasi. Pengujian independensi jika data yang digunakan meruppakan kategorik dan numerik dapat menggunakan uji *polyserial correlation.* Hipotesis yang dimiliki sama dengan hipotesis pada uji *pearson correlation,* namun untuk statistik ujinya berbeda seperti yang ditunjukkan pada persamaan (2.8) berikut.

$$r = \left(\frac{\hat{\rho}}{SE(\hat{\rho})}\right)^2, \qquad (2.8)$$

yang mana simbol $\hat{\rho}$ merupakan nilai *polyserial correlation*, $SE(\hat{\rho})$ merupakan nilai standar error dari $\hat{\rho}$, serta nilai r

merupakan nilai *polyserial correlation.* Untuk daerah penolakannya, jika nilai *r* lebih besar dari $\chi^2_{(\alpha,n)}$ maka keputusan yang diambil adalah tolak H_0 (SAS, 2019).

2.4.2 Estimasi Parameter

Estimasi parameter dilakukan dengan metode *Maximum Likelihood*. Metode ini memberikan landasan dalam mendekati nilai estimasi pada regresi logistik. Setiap pengamatan pada regresi logistk ini mengikuti distribusi *Bernoulli* sehingga fungsi *likelihood* dapat ditentukan. Jika variabel respon memiliki kategori 0 dan 1 dimana untuk variabel *y = 1* maka fungsi *likelihood* adalah $\pi(x_i)$ dan untuk variabel *y = 0* maka fungsi *likelihood* adalah $1-\pi(x_i)$. Fungsi probabilitas untuk setiap pasangan (x_i, y_i) dapat dituliskan pada persamaan (2.9)

$$\pi(x_i)^{y_i}(1-\pi(x_i))^{1-y_i}. \qquad (2.9)$$

Setiap pasangan pengamatan diasumsikan saling bebas sehingga fungsi *likelihood* merupakan gabungan dari fungsi distribusi masing-masing pasangan dan dapat dituliskan pada persamaan (2.10)

$$l(\boldsymbol{\beta}) = \prod_{i=1}^{n} \pi(x_i)^{y_i}\left[1-\pi(x_i)\right]^{1-y_i}. \qquad (2.10)$$

Fungsi *maximum likelihood* yang digunakan untuk mengestimasi $\boldsymbol{\beta}$ pada persamaan (2.10) akan lebih mudah di maksimumkan dalam bentuk persamaan. $\boldsymbol{\beta}$ merupakan vektor yang berisi nilai β . Persamaan untuk *log likelihood* dapat dituliskan pada persamaan (2.11)

$$L(\boldsymbol{\beta}) = \ln\left[l(\boldsymbol{\beta})\right] = \sum_{i=1}^{n}\left\{y_i \ln\left[\pi(x_i)\right] + (1-y_i)\ln\left[1-\pi(x_i)\right]\right\}. \qquad (2.11)$$

Untuk mencari nilai $\boldsymbol{\beta}$ yang memaksimum $L(\boldsymbol{\beta})$ maka $L(\boldsymbol{\beta})$ diturunkan terhadap β_0 dan β_1 lalu disamadengankan nol sehingga didapatkan persamaan *likelihood* pada persamaan (2.12) dan (2.13)

$$\sum [y_i - \pi(x_i)] = 0\,, \tag{2.12}$$

dan

$$\sum x_i\,[y_i - \pi(x_i)] = 0\,. \tag{2.13}$$

Nilai $\boldsymbol{\beta}$ yang didapatkan pada persamaan (2.12) dan (2.13) adalah hasil dari *maximum likelihood estimation* yang mengahasilkan nilai $\hat{\boldsymbol{\beta}}$ sehingga rumus yang dapat digunakan untuk memprediksi nilai pada regresi logistik ini dapat dituliskan dengan persamaan (2.14)

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \hat{\pi}(x_i)\,, \tag{2.14}$$

rumus tersebut menunjukkan bahwa total dari semua pengamatan (*y*) sama dengan total dari nilai prediksi (Hosmer & Stanley, 2000).

2.4.3 Uji Serentak

Uji secara simultan pada regresi logistik biner memiliki pada hipotesis sebagai berikut.

H_0: $\beta_1 = \beta_2 = \ldots = \beta_j = 0$ (tidak ada pengaruh antara variabel prediktor terhadap variabel respon)

H_1: minimal ada satu $\beta_j \neq 0$ (terdapat pengaruh antara variabel prediktor terhadap variabel respon)

Statistik uji yang digunakan adalah *G* yang mengikuti distribusi dengan derajat bebas sama dengan banyaknya parameter, dimana H_0 akan ditolak jika nilai statistik uji *G* akan lebih dari sama dengan nilai $\chi^2_{(p,\alpha)}$ dengan tingkat kepercayaan α. Rumus untuk statistik uji *G* dapat ditulisakan pada persamaan (2.15) (Hosmer & Stanley, 2000)

$$G = 2\left\{\begin{array}{l}\displaystyle\sum_{i=1}^{n}[y_i \ln(\hat{\pi}_i) + (1-y_i)\ln(1-\hat{\pi}_i)] \\ -[n_1 \ln(n_1) + n_0 \ln(n_0) - n\ln(n)\end{array}\right\}, \tag{2.15}$$

dimana $\hat{\pi}_i$ merupakan peluang kejadian sukses dengan nilai probabilitas antara 0 sampai 1, n_1 merupakan jumlah data dari variabel respon yang berkategori 1, n_0 merupakan jumlah data dari variabel respon yang berkategori 0 serta n adalah jumlah keseluruhan data.

2.4.4 Uji Individu

Pengujian secara parsial dilandaskan pada hipotesis adalah sebagai berikut.

H_0: $\beta_j = 0$ (tidak terdapat pengaruh antara masing-masing variabel prediktor terhadap variabel respon)

H_1: $\beta_j \neq 0$ (terdapat pengaruh antara masing-masing variabel prediktor terhadap variabel respon)

H_0 ditolak jika memiliki nilai P *value* yang kurang dari taraf signifikansi. Rumus statistik uji *Wald* dapat dituliskan pada persamaan (2.16) (Hosmer & Stanley, 2000)

$$W = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}, \tag{2.16}$$

dimana $\hat{\beta}_j$ merupakan koefisien parameter yang ke-*j* serta $SE(\hat{\beta}_j)$ adalah standar error dari koefisien parameter yang ke-*j*.

2.5 *Naïve Bayes Classifier*

Klasifikasi Bayes merupakan klasifikasi statistik yang digunakan untuk melakukan prediksi pada kelas suatu anggota probabilitas. Ide awal dari *Naive Bayes* ini yaitu dengan join probabilitas kata dan kategori yang diberikan oleh sebuah dokumen (Durajati & Gumelar, 2012). *Naive Bayes Classifier* memiliki dasar teori *Naïve Bayes* yang ditunjukkan pada persamaan (2.17).

$$P(\mathrm{Y} \mid \mathrm{X}) = \frac{P(\mathrm{X} \mid \mathrm{Y})\,\mathrm{P(Y)}}{P(X)}, \tag{2.17}$$

$P(X|Y)$ adalah suatu probabilitas untuk memperoleh nilai X sebagai input ketika sudah diketahui masuk pada kelas Y_i. Nilai $P(X)$ adalah suatu konstanta yang mempunyai nilai konstan dan dapat dihilangkan karena jika nilai probabilitas diurutkan berdasarkan kelas, maka nilai $P(X)$ akan selalu sama. Selanjutnya persamaan dapat disederhanakan menjadi persamaan (2.18)

$$P(\mathrm{Y} \mid \mathrm{X}) \propto P(\mathrm{X} \mid \mathrm{Y})\,\mathrm{P}(\mathrm{Y})\,. \qquad (2.18)$$

Ketika dilakukan klasifikasi, algoritma akan mencari nilai probabilitas paling tinggi dari semua kategori dokumen yang diujikan (V_{MAP}). Persamaan V_{MAP} dapat ditunjukkan pada persamaan (2.19)

$$V_{MAP} = \arg\max_{V_j \in V} P(V_j \mid x_1, x_2, \ldots x_k)\,, \qquad (2.19)$$

dimana x_k merupakan kata kunci ke-k dan dengan menerapkan teorema *Naive Bayes* pada persamaan (2.17), maka persamaan (2.19) dapat ditulis menjadi persamaan (2.20)

$$V_{MAP} = \arg\max_{V_j \in V} \frac{P(V_j)P(x_1, x_2, \ldots x_k \mid \mathrm{V}_j)}{P(x_1, x_2, \ldots x_k)}, \qquad (2.20)$$

karena nilai $P(x_1, x_2, \ldots, x_k)$ untuk semua nilai V_j sama berarti nilai V_j dapat diabaikan, maka persamaan (2.20) dapat ditulis menjadi persamaan (2.21)

$$V_{MAP} \propto \arg\max_{V_j \in V} P(V_j)P(x_1, x_2, \ldots x_k \mid \mathrm{V}_j)\,. \qquad (2.21)$$

Setiap kata dalam (x_1, x_2, ..., x_k) diasumsikan bersifat independen sehingga selanjutnya dapat diberikan persamaan (2.22)

$$\mathrm{P}(x_1, x_2, \ldots x_k \mid \mathrm{V}_j) = \prod_{k=1}^{n} P(x_k \mid \mathrm{V}_j)\,, \qquad (2.22)$$

maka akan didapatkan pendekatan yang dipakai dalam *Naïve Bayes Classifier* pada persamaan (2.23).

$$V_{MAP} = \arg\max_{V_j \in V} P(V_j) \prod_{k=1}^{n} P\left(x_k \mid \mathrm{V}_j\right) \qquad (2.23)$$

dimana, V_{MAP} merupakan nilai *output* hasil klasifikasi *Naïve Bayes*. Nilai $P(V_j)$ dapat dihitung dengan rumus pada persamaan (2.24),

$$P(V_j) = \frac{|doc_j|}{|training|} \tag{2.24}$$

dimana, $|doc_j|$ adalah jumlah *tweet* yang memiliki kategori *j* dalam *training*. Untuk $|training|$ adalah jumlah *tweet* dalam contoh yang digunakan *training*. Untuk setiap probabilitas kata x_k dalam setiap kategori dihitung pada saat *training* dengan persamaan (2.25),

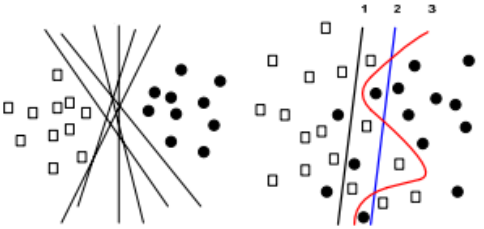
$$P(x_k \mid V_j) = \frac{n_k + 1}{|n + \text{kosakata}|} \tag{2.25}$$

n_k merupakan jumlah kemunculan pada kata x_k dalam *tweet* berkategori v_j sedangkan *n* adalah banyaknya seluruh kata dalam *tweet* dengan kategori V_j dan |kosakata| merupakan banyaknya kata dalam *training.*

2.6 *Support Vector Machine*

Support *Vector Machine* (SVM) pertama kali diperkenalkan pada akhir 1979 oleh Vapnik yang selanjutnya mendapatkan perhatian yang lebih pada tahun 1992 bersama dengan Boser (Hardle, Prastyo, & Hafner, 2013). *Support Vector Machine* adalah salah satu metode dalam *machine learning* yang digunakan untuk melakukan suatu prediksi dan pengklasifikasian. Ide dasar dalam *Support Vector Machine* merupakan gabungan dari beberapa konsep yang pernah ada sebelumnya dalam mengatasi permasalah terutama pada kasus klasifikasi dan prediksi. Konsep klasifikasi dengan menggunakan metode *Support Vector Machine* memiliki ide dasar bahwa menemukan fungsi pemisah (*hyperlane*) yang nantinya dapat memisahkan dua kelas secara optimal (Tan, Steinbach, Karpatne, & Kumar, 2018). Persamaan *hyperplane* dikatakan baik jika memiliki margin terbesar. Margin adalah dua

kali jarak antara *hyperplane* dan *support vector*, dimana *support vector* adalah titik yang berada paling dekat dengan *hyperplane.* Pada dasarnya SVM dikembangkan dengan prinsip *linier classifier* yang dapat dibedakan menjadi dua yaitu *linear separable* dan non *separable* sebagaimana pada Gambar 2.1.



Gambar 2.1 *Linear separable* (kiri), *Linear non-separable* (kanan)
(Sumber: (Hardle, Prastyo, & Hafner, 2013))

2.6.1 *Support Vector Machine Linear*

Support Vector Machine Linear adalah peneparan metode SVM pada data yang dipisahkan secara linier. Diberikan data yakni *D* adalah $(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_{|D|}, y_{|D|})$ dimana $\mathbf{X}_i$ adalah data *training* dengan label kelas y_i. Dimana y_i bernilai +1 atau -1. *Separating hyperlane* dapat dituliskan pada persamaan (2.26)

$$\mathbf{W}^{\mathrm{T}}\mathbf{X} + b = 0\,, \tag{2.26}$$

dimana $\mathbf{W} = \{w_1, w_2, \ldots, w_n\}$ yang merupakan nilai bobot, *n* adalah jumlah atribut, dan *b* adalah skalar. Jika x_1 dan x_2 adalah nilai dari suatu atribut untuk $\mathbf{X}$, dengan *b* adalah bobot tambahan maka dapat dituliskan persamaan (2.27)

$$w_0 + w_1 x_1 + w_2 x_2 = 0, \tag{2.27}$$

oleh karena itu, setiap nilai yang berada diatas *separating hyperlane* memiliki persamaan yang dituliskan pada persamaan (2.28)

$$w_0 + w_1 x_1 + w_2 x_2 > 0. \tag{2.28}$$

Selain itu, setiap nilai yang berada dibawah *separating hyperlane* memiliki persamaan yang dituliskan pada persamaan (2.29)

$$w_0 + w_1 x_1 + w_2 x_2 < 0. \quad (2.29)$$

Bobot yang diberikan dapat disesuaikan sehingga *hyperlane* dapat menentukan sisi dari margin yang dapat dituliskan pada persamaan (2.30) dan (2.31)

$$H_1 : w_0 + w_1 x_1 + w_2 x_2 \geq 1 \quad \text{for } y_i = +1, \quad (2.30)$$

$$H_2 : w_0 + w_1 x_1 + w_2 x_2 \leq -1 \quad \text{for } y_i = -1. \quad (2.31)$$

Jika nilai jatuh pada atau diatas H_1 maka akan masuk kelas +1 sedangkan jika nilai jatuh pada atau dibawah nilai H_2 maka akan masuk kelas -1. Persamaan (2.30) dan (2.31) dapat disamaakan dan didapatkan hasil pada persamaan (2.32)

$$y_i (w_0 + w_1 x_1 + w_2 x_2) \geq 1. \quad (2.32)$$

Berdasarkan hal tersebut, dapat ditentukan nilai margin yang maksimal. Jarak dari *separating hyperlane* untuk nilai pada setiap H_1 adalah $1/\|\mathbf{W}\|$, dimana $\|\mathbf{W}\|$ adalah jarak *Euclidean* dari $\mathbf{W} = \sqrt{\mathbf{W}\mathbf{W}}^2$. Untuk nilai jarak *separating hyperlane* pada setiap H_2 adalah margin maksimal, yakni $2/\|\mathbf{W}\|$.

2.6.2 *Support Vector Machine Non Linear*

Data yang digunakan biasanya tidak akan selalu bersifat *linear* separable, oleh karena itu susah untuk mendapatkan data yang *linear hyperlane*. Masalah ini dapat diatasi dengan mentransformasikan data ke bidang dengan dimensi lebih tinggi. *Support Vector Machine* dapat digunakan pada data non linier. Pada dasarnya, klasifikasi non linier dapat menyelesaikan masalah secara optimum dengan menggunakan persamaan pada (2.33)

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j). \quad (2.33)$$

$K(x_i, x_j)$ adalah fungsi kernel yang menunjukkan pola non linier pada bidang fitur (Sain & Purnami, 2015). Fungsi kernel yang banyak digunakan adalah sebagai berikut.

1. *Linear kernel* = $(x^T x)$
2. *Polynomial kernel* = $(x^T x + 1)^p$
3. *Gaussian radical-basic function* (RBF) = $K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2)$

2.6.3 Pemilihan Parameter Pada Fungsi Kernel SVM

Tuning parameter adalah langkah awal sebelum melakukan klasifikasi dengan SVM. Biasanya *tuning parameter* diketahui sebagai optimasi parameter. *Tuning parameter* dilakukan melalui *Grid search. Grid search* adalah suatu metode untuk menemukan *hyperplane* yang optimal pada model dengan hasil prediksi yang akurat. Pengoptimalan *hyperlane* dicari dengan menentukan parameter *Cost* senilai 0,01; 0,1; 1; 10; dan 100 untuk SVM dengan kernel linier dan parameter *Cost* senilai 0,01; 0,1; 1; 10; dan 100 serta Gamma senilai 0,01; 0,1; 1; 10; dan 100 untuk SVM dengan kernel RBF (Brownlee, 2019).

Analisis klasifikasi menggunakan metode *Grid search* SVM pada masing-masing data memiliki langkah-langkah yakni dimulai dengan menentukan *range* nilai C dan Gamma. Lalu dilakukan klasifikasi SVM dengan kombinasi nilai parameter C dan Gamma. Untuk selanjutnya dihitung nilai akurasinya. Apabila terdapat kombinasi nilai parameter C dan Gamma yang belum dilakukan, maka kembali lagi ke langkah melakukan klasifikasi SVM dengan kombinasi C dan Gamma. Jika semua kombinasi sudah dilakukan makan dilanjutkan dengan menentukan nilai parameter C dan Gamma yang paling optimal dari seluruh kombinasi parameter yang sudah dilakukan.

2.7 *K-fold Cross Validation*

Metode *k-fold cross validation* adalah suatu metode yang diketahui untuk memprediksi tingkat kesalahan dalam teknik klasifikasi. Metode ini banyak digunakan untuk mengurangi bias yang berhubungan dengan data random. *K-fold cross validati-on* membagi data kedalam beberapa bagian (*subset)* yang mana disebut *fold* (Gokgoz & Subasi, 2015). Data secara random dipartisi menjadi *k* himpunan *fold* sehingga dapat ditulis menjadi $D_1, D_2, \ldots, D_k$, dimana setiap bagian memiliki ukuran yang sama. Pada metode ini data dibagi menjadi data *training* dan data *testing*. Dilakukan pengulangan sebanyak *k* kali dalam data *training* dan data *testing*. Pada iterasi *i,* partisi D_i digunakan sebagai data *testing*, dan sisanya digunakan sebagai data *training*. Sebagai contoh, bagian $D_2, \ldots, D_k$ digunakan untuk data *training* pada model pertama dimana D_1 merupakan data *testing* dan seterusnya. Untuk klasifikasi, estimasi akurasi yang diperoleh adalah sebesar jumlah yang diklasifikasikan benar dari *k* iterasi (Han, Kamber, & Pei, 2012).

2.8 *Synthetic Minority Oversampling Technique* (SMOTE)

Pada jumlah data yang tidak seimbang atau *imbalance* terdapat perbedaan jumlah antara kategori yang sangat signifikan, yang mana terdapat kategori yang dominan dan disebut mayoritas serta adanya kategori yang lebih sedikit dan disebut minoritas. *Synthetic Minority Oversampling Technique* atau disingkat SMOTE adalah suatu metode algoritma yang dikenalkan oleh Nithes V Chawla pada tahun 2002 (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Pada metode ini diterapkan salah satu metode *oversampling* dimana dilakukan dengan menerapkan metode sampling. Untuk replikasi datanya, SMOTE melakukan dengan pemilihan *k nearest neighbor* sebagai penentuannya, sehingga nantinya jumlah data yang minoritas akan menjadi seimbang dengan data mayoritas. Pada metode SMOTE ini dicari *k nearest neighbour* untuk setiap data pada kategori minoritas yang

selanjutnya akan dibuat data sintesis sebanyak persentase replikasi data minoritas dengan rumus pada persamaan (2.34) berikut.

$$N\% = \frac{\text{Jumlah data mayoritas}}{\text{Jumlah data minoritas}} \times 100\% \ . \qquad (2.43)$$

Untuk selanjutnya akan dihitung nilai *nearest neighbour* yang dipilih berdasarkan pada jarak Euclidian dari kedua data. Rumus untuk jarak Euclidian dapat ditulisakan pada persamaan (2.35) berikut.

$$x_{knn} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_k - y_k)^2} \ . \qquad (2.35)$$

Secara umum rumus untuk menghitung data sintesis dapat dirumuskan pada persamaan (2.36) berikut.

$$x_{syn} = x_i + (x_{knn} - x_i)\delta \ , \qquad (2.36)$$

dimana, x_{syn} merupakan data sintesis hasil replikasi, x_i merupakan data yang akan direplikasi, x_{knn} adalah data yang memiliki nilai jarak paling dekat dengan data yang akan direplikasi, δ merupakan angka acak antara 0 hingga 1, dan *i* merupakan *tweet* ke 1, 2, …, n . Untuk lebih jelasnya akan dilakukan ilustrasi penerapan SMOTE dengan menggunakan 10 sampel data awal Bank BRI yang disajikan pada Tabel 2.1 berikut.

Tabel 2.1. Sampel untuk SMOTE

Tweet	Klasifikasi	bisa	cek	kredit
1	0	0.3320018	0.2513897	0.3228815
2	0	0.1776388	0.269014	0.3455178
3	0	0.2153781	0	0
4	1	0.1715825	0.1932975	0.2804389
5	1	0	0	0
6	1	0.245249	0	0
7	1	0	0	0.3065184
8	1	0	0.4468154	0
9	1	0.2711522	0.4810029	0
10	1	0	0	0.2288338

Berdasarkan pada Tabel 2.1 maka dapat diketahui bahwa data yang akan dilakukan replikasi adalah data dari kelas minoritas

yakni klasifikasi dengan kategori 0. Jumlah klasifikasi dengan kategori 0 terdapat sebanyak 3 data sedangkan klasifikasi dengan kategori 1 terdapat 7 data sehingga nilai persentase SMOTE yang akan digunakan dapat dihitung sebagai berikut

$$N\% = \frac{7}{3} \times 100\% = 233,3\% .$$

Selanjutnya adalah menentukan data yang memiliki nilai jarak paling dekat dengan data yang akan direplikasi yang pada ilustrasi ini dimulai dengan perhitungan antara *tweet* 1 dengan *tweet* 2 dibandingkan dengan *tweet* 1 dengan *tweet* 3.

Tweet 1 dengan *tweet* 2

$$d\left(\begin{bmatrix}0,332\\0,251\\0,322\end{bmatrix},\begin{bmatrix}0,177\\0,269\\0,345\end{bmatrix}\right) = \sqrt{(0,332-0,177)^2 + \ldots + (0,322-0,345)^2} = 0,157$$

Tweet 1 dengan *tweet* 3

$$d\left(\begin{bmatrix}0,332\\0,251\\0,322\end{bmatrix},\begin{bmatrix}0,215\\0\\0\end{bmatrix}\right) = \sqrt{(0,332-0,215)^2 + \ldots + (0,322-0)^2} = 0,425$$

Data *tweet* yang memiliki jarak paling dekat berdasarkan perhitungan terdapat pada *tweet* 1 dengan *tweet* 2 sehingga x_{knn} yang digunakan adalah pada *tweet* 2. Untuk perhitungan data sintesis berdasarkan persamaan (2.34) pada ilustrasi ini adalah sebagai berikut dengan menggunakan nilai δ sebesar 0,5.

$$x_{syn} = \begin{bmatrix}0,332\\0,251\\0,322\end{bmatrix} + \left(\begin{bmatrix}0,177\\0,269\\0,345\end{bmatrix} - \begin{bmatrix}0,332\\0,251\\0,322\end{bmatrix}\right) \times 0,5 = \begin{bmatrix}0,254\\0,260\\0,334\end{bmatrix}$$

Berdasarkan hasil tersebut dapat diketahui bahwa data sintesis yang diperoleh untuk replikasi pertama yakni senilai bri_{syn}=0,254; cek_{syn}=0,260; $kredit_{syn}$=0,334; dan $klasifikasi_{syn}$=0 dan seterusnya untuk observasi yang direplikasi sampai sebanyak persentase *oversampling* yang telah dihitung sebelumnya.

2.9 Ketepatan Klasifikasi

Pengukuran ketepatan klasifikasi dilakukan untuk melihat performa klasifikasi yang telah dilakukan. Dalam mengukur ketepatan klasifikasi, perlu diketahui jumlah pada setiap kelas prediksi dan kelas aktual yang terdiri dari *TP* (*True Positif*) yaitu jumlah *tweet* bersentimen positif yang tepat diprediksi dalam kelas positif, *TN* (*True Negatif*) yaitu *tweet* yang tepat terprediksi dalam kelas negatif, *FP* (*False Positif*) yaitu *tweet* bersentimen negatif yang terprediksi dalam kelas positif, dan *FN* (*False Negatif*) yaitu *tweet* bersentimen positif yang terprediksi dalam kelas negatif (Bekkar, Djemaa, & Alitouche, 2013).

Tabel 2.2. *Confusion Matrix*

Kelas Aktual	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	*TP*	*FN*
Negatif	*FP*	*TN*

Tabel 2.2 menunjukkan bahwa *confusion matrix* digunakan untuk menghitung ketepatan klasifikasi. Nilai yang berada pada diagonal utama menunjukkan hasil keputusan yang benar. Berikut adalah beberapa kriteria untuk menilai ketepatan klasifikasi yang disajikan pada persamaan (2.37), persamaan (2.38), dan persamaan (2.39)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FP} \ , \qquad (2.37)$$

$$precision = \frac{TP}{TP + FP} \ , \qquad (2.38)$$

$$recall = \frac{TP}{TP + FN} \ . \qquad (2.39)$$

Secara grafik, *Receiver Operating Characteristic* atau disebut ROC sering dihasilkan dalam kurva yang memiliki jumlah sentimen positif sebagai fungsi dari sentimen yang bersifat negatif tetapi diprediksi pada sentimen positif. Secara khusus, ROC

mendekati hasil sebagai fungsi dari (*1-specifity*) untuk semua kemungkinan batas nilai. Untuk *Area Under Curve* (AUC) dapat mengurutkan model dari seluruh model yang ada. AUC dapat mengestimasi dengan berbagai teknik, salah satunya adalah metode trapezoidal yang merupakan metode geomterik yang berdasarkan pada interpolasi linier antara nilai pada ROC. Beberapa peneliti menemukan sebuah rumus yang dapat mendekati nilai AUC dengan menggunakan persamaan (2.40)

$$AUC = \frac{1}{2}(sensitivity + specificity)\,. \qquad (2.40)$$

Nilai AUC sama dengan nilai peluang dari hasil klasifikasi yang akan memberikan peringkat *random positif test case* lebih tinggi daripada *random negatif test.* Nilai AUC berada diantara 0,5 hingga 1 (Bekkar, Djemaa, & Alitouche, 2013). Tabel 2.3 berikut akan menjelaskan kebaikan dari nilai AUC.

Tabel 2.3. Kebaikan Nilai AUC

Nilai AUC	Kebaikan Model
0,5 – 0,6	Kurang
0,6 – 0,7	Cukup
0,7 – 0,8	Baik
0,8 – 0,9	Sangat Baik
0,9 – 1,0	Sempurna

2.10 *Word Cloud*

Word Cloud merupakan suatu metode untuk membuat visualisasi data yang berasal dari dokumen atau teks. *Word cloud* adalah perwakilan grafis dari sebuah teks atau dokumen dengan cara melakukan *plotting* pada kata-kata yang mempunyai frekuensi kemunculan paling banyak dan digambarkan pada ruang berdimensi dua. Kata yang memiliki frekuensi sering muncul dapat ditunjukkan melalui ukuran dari huruf kata tersebut. Berikut adalah visualisasi dari *word cloud* yang disajikan pada Gambar 2.2 (Jin, 2016).

Gambar 2.2 Visualisasi *Word Cloud* (Sumber: (Jin, 2016))

2.11 Bank Rakyat Indonesia (BRI)

Bank Rakyat Indonesia (BRI) adalah salah satu bank milik pemerintah yang terbesar di Indonesia. Pada awalnya BRI didirikan di Purwokerto, Jawa Tengah. Lembaga tersebut berdiri tanggal 16 Desember 1895, yang kemudian dijadikan sebagai hari kelahiran BRI. Sejak 1 Agustus 1992 berdasarkan Undang-Undang Perbankan No. 7 tahun 1992 dan Peraturan Pemerintah RI No 21 tahun 1992 status BRI berubah menjadi perseroan terbatas (BRI, 2019). Pada saat ini, BRI masuk kedalam 2000 perusahaan terbesar dunia yang dirilis oleh *Forbes* pada tahun 2018. BRI mendapat peringkat 415 yang menjadikan BRI adalah perusahaan dengan peringkat terbaik diantara perusahaan-perusahaan yang ada di Indonesia (Forbes, 2018). Pada saat ini selain melalui layanan *call center*, BRI juga aktif dalam berbagai sosial media. Salah satu akun media sosial yang digunakan adalah Twitter, dengan nama akun @kontakBRI yang digunakan para nasabah untuk mengungkapkan pendapat atau mengadu mengenai pelayan BRI.

2.12 Bank Mandiri

Bank Mandiri didirikan pada 2 Oktober 1998, sebagai bagian dari program restrukturisasi perbankan yang dilaksanakan oleh pemerintah Indonesia. Pada bulan Juli 1999, empat bank pemerintah yaitu Bank Bumi Daya, Bank Dagang Negara, Bank Ekspor Impor Indonesia dan Bank Pembangunan Indonesia dilebur

menjadi Bank Mandiri, dimana masing-masing bank tersebut memiliki peran yang tak terpisahkan dalam pembangunan perekonomian Indonesia. Hingga Maret 2018, jaringan Bank Mandiri telah tersebar di seluruh Indonesia yang meliputi 4.948 jaringan kantor, meliputi 2.633 cabang dan 2.315 jaringan mikro. Layanan distribusi Bank Mandiri juga dilengkapi dengan 17.429 unit ATM yang terhubung dalam jaringan ATM Link, ATM Bersama, ATM Prima dan Visa/Plus, *Electronic Data Capture* (EDC) serta jaringan *e-banking* yang meliputi Mandiri Online, SMS Banking dan Call Center 14.000. Pada saat ini, Bank Mandiri masuk kedalam 2000 perusahaan terbesar di dunia yang dirilis oleh *Forbes* pada tahun 2018. Bank Mandiri mendapat peringkat 489 yang menjadikan Bank Mandiri adalah perusahaan dengan ranking kedua terbaik diantara perusahaan-perusahaan yang ada di di Indonesia (Forbes, 2018). Pada saat ini Bank Mandiri fokus pada digitalisasi setiap layanannya. Untuk pengaduan selain melalui layanan *call center*, Bank Mandiri juga aktif dalam berbagai sosial media antara lain Twitter, Facebook, dan Instagram. Salah satu akun media sosial yang dimiliki yakni Twitter, dengan nama akun @mandiricare yang digunakan para nasabah untuk mengungkapkan pendapat atau mengadu mengenai pelayanan Bank Mandiri. Melalui Mandiri 4.0, Bank Mandiri menginginkan setiap orang dapat melakukan transaksi perbankan di mana pun dan kapan pun termasuk untuk pembayaran *e-commerce* atau pun ke banyak *merchant.*

Halaman ini sengaja dikosongkan

BAB III
METODOLOGI PENELITIAN

Pada bab metodologi penelitian ini dijelaskan dari mana sumber data diperoleh, variabel apa saja yang digunakan untuk penelitian, struktur data penelitian, langkah analisis, serta diagram alir untuk penelitian ini.

3.1 Sumber Data

Sumber data yang digunakan adalah data sekunder. Data sekunder diambil dari kumpulan *tweet* para pengguna Twitter di Indonesia. Akun Twitter yang digunakan pada penelitian ini adalah @kontakBRI dan akun Twitter @mandiricare. Data *tweet* diambil mulai tanggal 7 Februari 2019 hingga 8 April 2019 dengan menggunakan Twitter API.

3.2 Variabel Penelitian dan Struktur Data

Variabel penelitian yang digunakan dalam penelitian ini terdiri dari variabel respon (Y) dan variabel prediktor (X) yang disajikan pada Tabel 3.1 berikut.

Tabel 3.1. Variabel Penelitian

Variabel	Nama Variabel	Skala Data
Y	Sentimen 0 = Sentimen positif 1 = Sentimen negatif	Nominal
X_k	Bobot kata kunci dari setiap *tweet* yang diperoleh dari hasil *tf-idf*	Rasio

Berikut ini adalah struktur data yang digunakan dalam penelitian ini setelah dilakukan *text preprocessing* yang disajikan pada Tabel 3.2. Struktur data yang digunakan adalah x_{ijk}, dimana i=1, 2 yang merupakan variabel bank yang digunakan yakni terdiri dari Bank BRI dan Bank Mandiri, j=1, 2, …, n yang merupakan banyaknya *tweet* pada setiap variabel

bank, dan *k*=1, 2, m yang merupakan kata kunci dari *tweet* ke-*j* dan bank ke-*i*.

Tabel 3.2. Struktur Data

Bank	**Tweet**	**Klasifikasi sentimen (y)**	**Kata Kunci (x_1)**	**Kata Kunci (x_2)**	⋯	**Kata Kunci (x_k)**
BRI (1)	1	$y_{1,1}$	$x_{11,1}$	$x_{11,2}$	⋯	$x_{11,k}$
	2	$y_{1,2}$	$x_{12,1}$	$x_{12,2}$	⋯	$x_{12,k}$
	3	$y_{1,3}$	$x_{13,1}$	$x_{13,2}$	⋯	$x_{13,k}$
	⋮	⋮	⋮	⋮	⋯	⋮
	n_1	$y_{1,n1}$	$x_{1n1,1}$	$x_{1n1,2}$	⋯	$x_{1n1,k}$
Bank Mandiri (2)	1	$y_{2,1}$	$x_{21,1}$	$x_{21,2}$	⋯	$x_{21,k}$
	2	$y_{2,2}$	$x_{22,1}$	$x_{22,2}$	⋯	$x_{22,k}$
	3	$y_{2,3}$	$x_{23,1}$	$x_{23,2}$	⋯	$x_{23,k}$
	⋮	⋮	⋮	⋮	⋯	⋮
	n_2	$y_{2,n2}$	$x_{2n2,1}$	$x_{2n2,2}$	⋯	$x_{2n2,k}$

3.3 Langkah Analisis

Langkah analisis yang digunakan untuk mencapai tujuan adalah sebagai berikut.

1. Mengambil *tweet* menggunakan Twitter API.
   a. Memasukkan *keyword* yang berhubungan dengan @kontakBRI dan akun Twitter @mandiricare.
   b. Menyimpan hasil *crawling data* dari kedua akun Twitter.
   c. Mengklasifikasikan *tweet* kedalam sentimen positif atau negatif dengan menggunakan *lexicon based*. Kamus yang

digunakan adalah kamus kata kunci sentimen positif dan kamus kata kunci sentimen negatif (Hu & Liu, 2004).

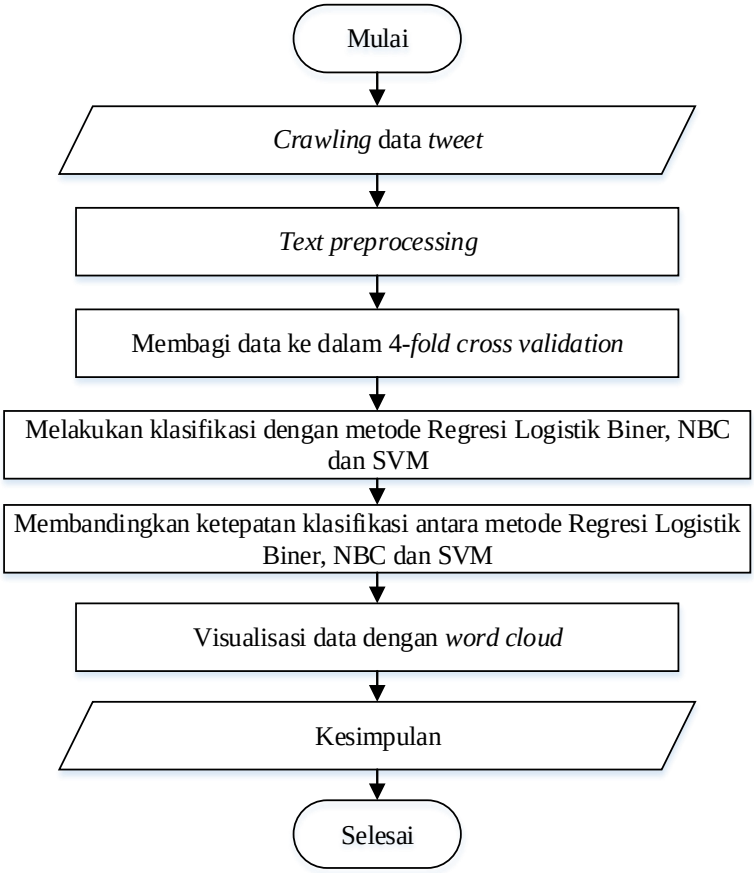
2. Melakukan *text preprocessing* yang meliputi :
   a. Melakukan *cleansing*, yakni menghapus *username*, *hastag*, RT, baris kosong, *punctuation*, spasi berlebih, dan *URL*,
   b. Melakukan *case folding*, yakni mengubah semua karakter menjadi huruf kecil,
   c. Melakukan *stemming*, yakni mengubah kata menjadi kata dasar,
   d. Melakukan *stopword*. Daftar *stopwords* diambil dari thesis yang ditulis oleh Fadilla. Z. Tala yang berjudul "*A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*",
3. Membagi data dalam 4-*fold cross validation.*
4. Melakukan klasifikasi menggunakan metode Regresi Logistik Biner
   a. Mengubah data *tweet* ke dalam bentuk frekuensi kemunculan kata menggunakan *tf-idf*.
   b. Melakukan uji signifikansi parameter secara individu dengan persamaan (2.14),
   c. Melakukan estimasi parameter,
   d. Menentukan model dari regresi logistik biner,
   e. Menghitung kebaikan dari model yang diperoleh berdasarkan parameter sebelumnya,
   f. Mengulang langkah a sampai f dengan menggunakan data yang telah diklasifikasikan dengan SMOTE dan data *lexicon*.
5. Melakukan klasifikasi menggunakan metode *Naïve Bayes Classifier*
   a. Mengubah data *tweet* ke dalam bentuk frekuensi kemunculan kata menggunakan *tf-idf*.
   b. Menghtiung probabilitas dari V_j pada data *training*, dimana V_j merupakan kategori sentimen yaitu V_1 = positif dan V_2 = negatif,

c. Menghitung probabilitas kata x_k pada kategori V_j
d. Model probabilitas *Naïve Bayes Classifier* disimpan dan digunakan pada data *testing*,
e. Menghitung probabilitas tertinggi dari kategori sentimen yang diujikan (V_{MAP}),
f. Mencari nilai V_{MAP} paling maksimum dan memasukkan *tweet* tersebut pada kategori dengan V_{MAP} maksimum,
g. Menghitung kebaikan dari model yang terbentuk,
h. Mengulang langkah a sampai h dengan menggunakan data yang telah diklasifikasikan dengan SMOTE dan data *lexicon*.

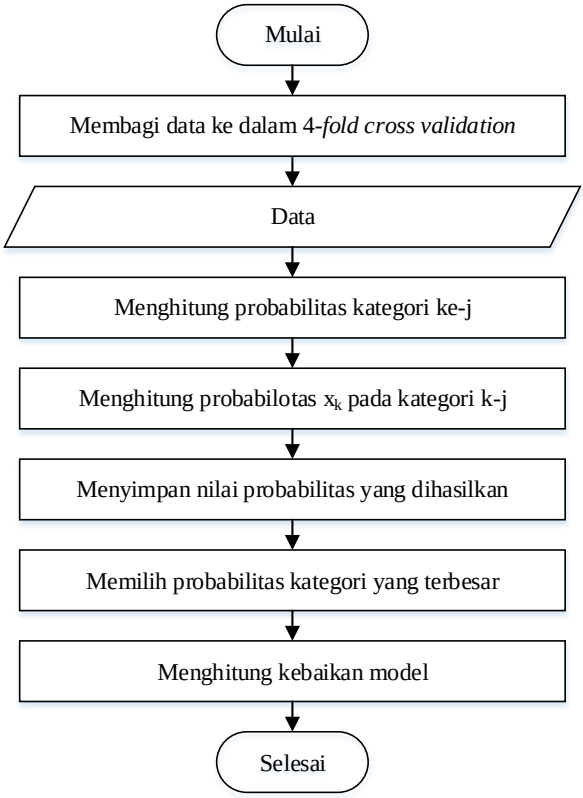
6. Melakukan klasifikasi menggunakan metode *Support Vector Machine*
   a. Mengubah data *tweet* ke dalam bentuk frekuensi kemunculan kata menggunakan *tf-idf*.
   b. Menentukan pembobot parameter pada SVM dengan jenis kernel *linear*,
   c. Menentukan parameter optimum pada SVM dengan jenis kernel *linear*,
   d. Membangun model SVM dengan jenis kernel *linear*,
   e. Menentukan pembobot parameter pada SVM dengan jenis kernel RBF,
   f. Menentukan parameter optimum pada SVM dengan jenis kernel RBF,
   g. Membangun model SVM dengan jenis kernel RBF,
   h. Memilih jenis kernel yang optimum berdasarkan hasil kebaikan model berdasarkan parameter yang didapat sebelumnya,
   i. Mengulang langkah a sampai h dengan menggunakan data yang telah diklasifikasikan dengan dengan SMOTE dan data *lexicon*.
7. Membandingkan hasil klasifikasi dari metode NBC, SVM, dan Regresi Logistik Biner dengan kriteria AUC.
8. Melakukan visualisasi dengan *word cloud.*
9. Menarik kesimpulan dan saran.

3.4 Diagram Alir

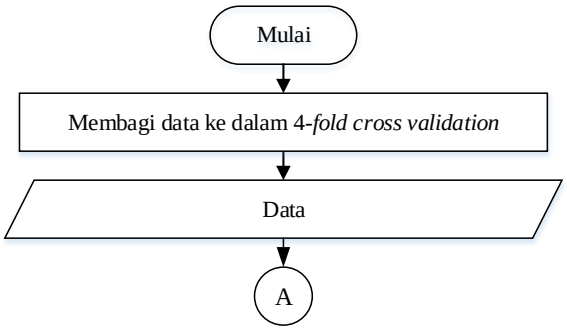
Diagram alir berdasarkan langkah analisis pada penelitian ini dapat digambarkan sebagai berikut.

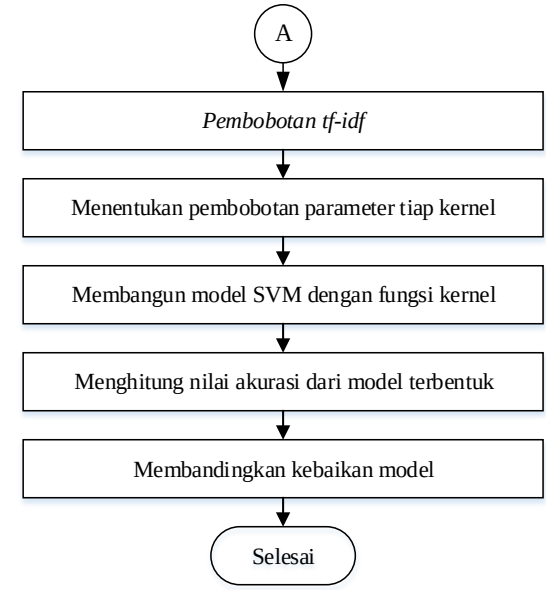


Gambar 3.1 Diagram Alir Penelitian.

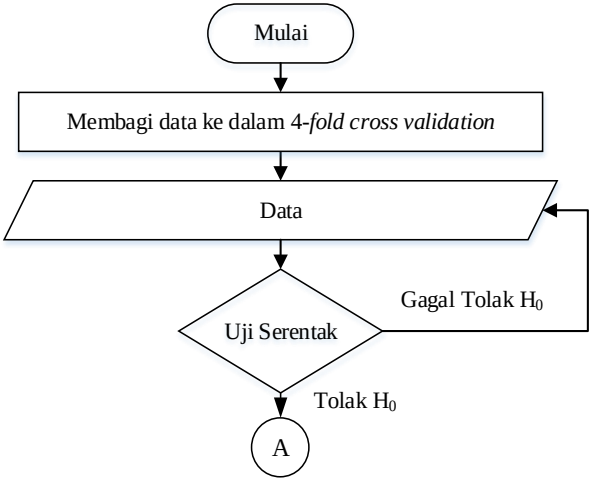

Mulai
Membagi data ke dalam 4-*fold cross validation*
Data
Menghitung probabilitas kategori ke-j
Menghitung probabilotas x_k pada kategori k-j
Menyimpan nilai probabilitas yang dihasilkan
Memilih probabilitas kategori yang terbesar
Menghitung kebaikan model
Selesai

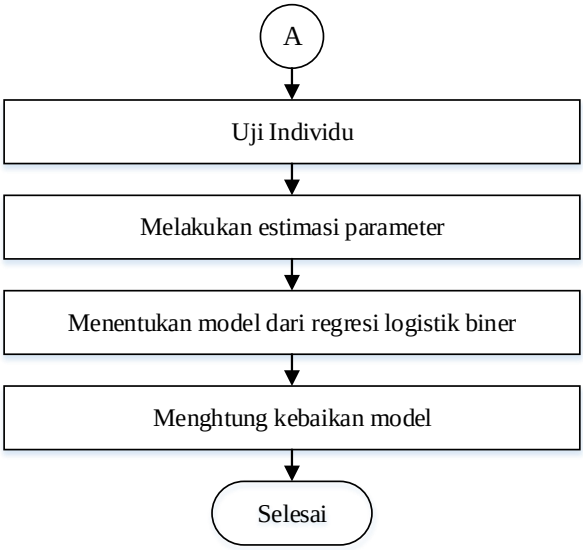
(a)


Mulai
Membagi data ke dalam 4-*fold cross validation*
Data
A


A
Pembobotan tf-idf
Menentukan pembobotan parameter tiap kernel
Membangun model SVM dengan fungsi kernel
Menghitung nilai akurasi dari model terbentuk
Membandingkan kebaikan model
Selesai

(b)


Mulai
Membagi data ke dalam 4-*fold cross validation*
Data
Uji Serentak
Gagal Tolak H_0
Tolak H_0
A



(c)

Gambar 3.2 Diagram Alir Klasifikasi NBC (a), SVM (b), dan Regresi Logistik Biner (c).

BAB IV
ANALISIS DAN PEMBAHASAN

Pada bab ini akan dijelasakan mengenai analisis dan pembahasan untuk mengklasifikasikan *tweet* tentang pelayanan nasabah pada akun Twitter BRI dan Bank Mandiri.

4.1 Praproses Teks

Tweet yang telah dikumpulkan kemudian dilakukan praproses teks, yaitu mulai dari menghapus *link*, menghapus *hastag*, menghapus simbol *retweet* (RT), menghapus *username*, menghapus angka, menghapus *emoticon*, menghapus baris kosong, menghapus *punctuation*, menghapus spasi berlebih, menghapus *underscore*, *case folding*, *stemming*, dan terakhir dilakukan *stopwords*.

Tabel 4.1. Contoh Praproses Teks

Praproses	Hasil
Tweet	@kontakBRI halo min, maaf ini BRI kenapa gangguan terus ya, susah mau transfer, sekalinya bisa malah keambil dua kali dananya.. Kan rugi saya,, mohon bantuannya..
Menghapus *username*	halo min, maaf ini BRI kenapa gangguan terus ya, susah mau transfer, sekalinya bisa malah keambil dua kali dananya.. Kan rugi saya,, mohon bantuannya..
Menghapus baris kosong	halo min, maaf ini BRI kenapa gangguan terus ya, susah mau transfer, sekalinya bisa malah keambil dua kali dananya.. Kan rugi saya,, mohon bantuannya..
Menghapus *punctuation*	halo min maaf ini BRI kenapa gangguan terus ya susah mau transfer sekalinya bisa malah keambil dua kali dananya Kan rugi saya mohon bantuannya
Menghapus spasi berlebih	halo min maaf ini BRI kenapa gangguan terus ya susah mau transfer sekalinya bisa malah keambil dua kali dananya Kan rugi saya mohon bantuannya

Tabel 4.1. Contoh Praproses Teks (Lanjutan)

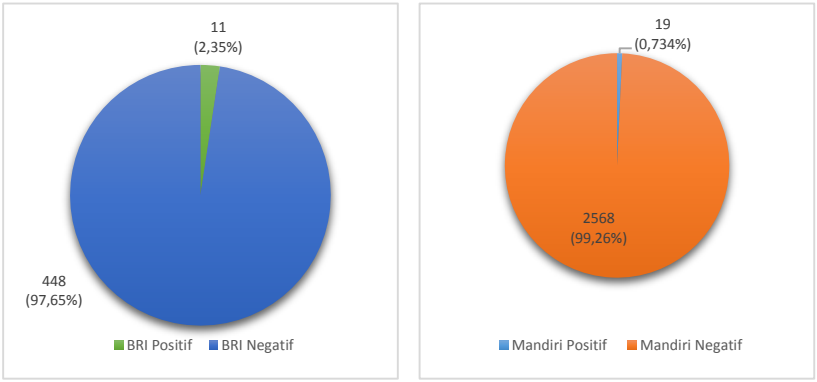
Praproses	Hasil
Case folding	halo min maaf ini bri kenapa gangguan terus ya susah mau transfer sekalinya bisa malah keambil dua kali dananya kan rugi saya mohon bantuannya
Stemming	'halo min maaf ini bri kenapa ganggu terus ya susah mau transfer sekali bisa malah ambil dua kali dana kan rugi saya mohon bantu'
Stopwords	['halo', 'maaf', 'bri', 'ganggu', 'susah', 'transfer', 'dana', 'rugi', 'mohon', 'bantu']

Tabel 4.1 menunjukkan bahwa sebuat *tweet* yang dipraposes dari awal hingga menjadi hasil akhir pada baris *stopwords.* Untuk daftar kata *stopwords* pada contoh praproses teks diambil dari thesis yang ditulis oleh Fadilla. Z. Tala yang berjudul "*A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*". Sehingga berdasarkan contoh praproses teks pada Tabel 4.1 dapat diketahui hasil akhir dari *stopwords* adalah berupa kata dasar yang selanjutnya kata dasar tersebut akan menjadi kata kunci-kata kunci yang telah dijelaskan pada Tabel 3.2 mengenai struktur data.

4.2 Karakteristik *Tweet*

Pada penelitian ini data yang digunakan yakni *tweet* pada Bank BRI yang mana setiap *tweet* tersebut mengandung suatu sentimen yang dibedakan menjadi sentimen positif dan sentimen negatif. Berikut adalah karakteristik dari perbandingan jumlah *tweet* bersentimen positif dan bersentimen negatif yang disajikan pada Gambar 4.1 untuk data dengan klasifikasi *tweet* pelayanan nasabah bank secara manual atau untuk selanjutnya disebut data awal. Frekuensi data *tweet* setiap sentimen pada Gambar 4.1 menunjukkan bahwa *tweet* dengan sentimen negatif memiliki jumlah lebih banyak daripada *tweet* dengan sentimen positif. Jumlah *tweet* yang mengandung sentimen negatif ada sebanyak 448 data dari 459 data *tweet* untuk BRI dan 2568 *tweet* dari 2587 untuk Bank Mandiri. Untuk *tweet* yang mengandung setimen

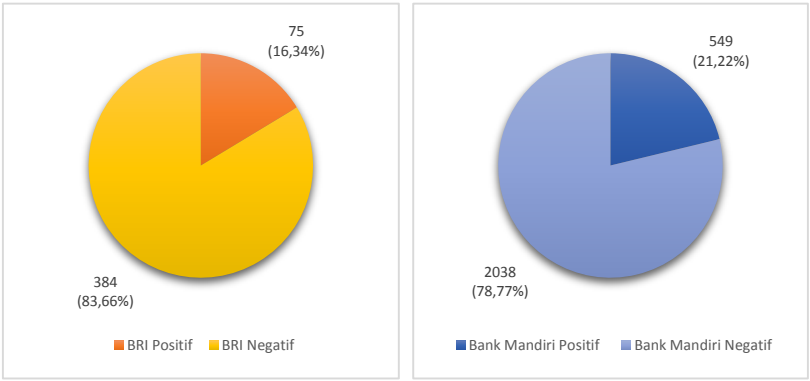
positif hanya ada sebanyak 11 data *tweet* untuk BRI dan 19 data *tweet* Bank Mandiri.



(a) (b)

Gambar 4.1 Frekuensi Data *Tweet* Setiap Sentimen Untuk Data dengan Klasifikasi Manual BRI (a), dan Bank Mandiri (b)

Hal ini menunjukkan bahwa jumlah *tweet* sentimen positif dan negatif pada Gambar 4.1 memiliki jumlah yang tidak seimbang dan menunjukkan bahwa banyak nasabah yang melakukan komplain melalui akun Twitter. Berikut karakteristik data *tweet* untuk data dengan klasifikasi menggunakan *lexicon*.



(a) (b)

Gambar 4.2 Frekuensi Data *Tweet* Setiap Sentimen Untuk Data dengan Klasifikasi *Lexicon* BRI (a), dan Bank Mandiri (b)

Gambar 4.2 menunjukkan bahwa *tweet* pada data klasifikasi dengan kamus *lexicon* memiliki sentimen negatif yang jumlahnya lebih banyak daripada *tweet* yang memiliki sentimen positif baik antara pada *tweet* BRI atau Bank Mandiri. Untuk lebih jelasnya dibuat perbandingan klasifikasi pada Tabel 4.2 dan 4.3 berikut.

Tabel 4.2. Perbandingan pada Data BRI

Klasifikasi Manual	*Lexicon*	
	Positif	Negatif
Positif	7	4
Negatif	68	380

Tabel 4.2 menunjukkan bahwa data positif yang dilakukan dengan klasifikasi manual diklasifikasikan benar positif dengan menggunakan *lexicon* adalah 7 data *tweet*. Untuk data klasifikasi manual positif yang salah diklasifikasikan pada data negative berdasarkan data *lexicon* ada sebanyak 4 data sedangkan ada 68 data positif yang salah diklasifikasikan menjadi negatif. Terdapat 380 data *tweet* negatif yang benar diklasifikasikan ke dalam *tweet* negatif.

Tabel 4.3. Perbandingan pada Data Bank Mandiri

Klasifikasi Manual	*Lexicon*	
	Positif	Negatif
Positif	11	8
Negatif	538	2030

Tabel 4.3 menunjukkan bahwa terdapat 11 data positif yang benar diklasifikasikan positif berdasarkan *lexicon,* namun terdapat 546 data yang salah diklasifikasikan dengan menggunakan *lexicon*. Terdapat 2030 data *tweet* negatif klasifikasi manual yang benar diklasifikasikan ke dalam *tweet* negatif dengan klasifikasi *lexicon.*

4.3 Klasifikasi pada data BRI

Pada klasifikasi dengan data BRI ini akan dilakukan beberapa metode klasifikasi yang digunakan untuk selanjutnya metode tersebut akan dibandingkan dan dipilih model dengan performa klasifikasi terbaik. Dalam penelitian ini digunakan 4-*fold*

cross validation. Untuk data SMOTE ini diterapkan salah satu metode *oversampling* dimana dilakukan dengan menerapkan metode sampling. Metode sampling pada SMOTE ini dilakukan untuk meningkatkan jumlah kelas minoritas dengan cara replikasi data secara acak. Untuk replikasi datanya, SMOTE melakukan dengan pemilihan *k nearest neighbor* sebagai penentuannya, sehingga nantinya jumlah data yang minoritas akan menjadi seimbang dengan data mayoritas. Untuk lebih jelasnya mengenai penarapan SMOTE telah diberikan ilustrasi perhitungan pada subbab 2.8 dengan menggunakan 10 sampel data acak yang diambil dari data awal Bank BRI.

4.3.1 TF-IDF untuk data BRI

Setelah dilakukan tahap *stopwords* maka langkah selanjutnya adalah akan dihitung bobot atau nilai TF-IDF dari setiap kata kunci yang muncul per *tweet*. Berikut adalah hasil perhitungan TF-IDF untuk data *tweet* pada akun BRI yang disajikan pada Tabel 4.4.

Tabel 4.4. TF-IDF pada Data BRI

Tweet	atm	gagal	ganggu	⋯	potong	rekening	rusak
1	0	0	0	⋯	0	0	0
2	0	0	0,356	⋯	0	0,360	0
3	0,324	0,161	0	⋯	0,177	0	0
4	0,221	0	0	⋯	0,242	0	0
5	0,181	0	0	⋯	0	0	0,346
6	0	0	0	⋯	0	0	0
7	0	0	0	⋯	0	0	0
⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮
453	0	0,259	0	⋯	0	0	0
454	0	0	0	⋯	0	0	0
455	0	0	0	⋯	0	0	0
456	0	0	0	⋯	0	0	0
457	0,1807	0,286	0	⋯	0	0	0
458	0	0	0,439	⋯	0	0	0
459	0	0,324	0	⋯	0	0	0

Data yang telah berbentuk TF-IDF seperti yang disajikan pada Tabel 4.4 tersebut akan digunakan untuk analisis lebih lanjut. Data *tweet* BRI memiliki jumlah kata kunci sebanyak 186 kata. Berikut ini akan disajikan frekuensi kemunculan kata tertinggi dari akun Twitter Bank BRI.

Tabel 4.5. Frekuensi Kemunculan Kata pada *Tweet* Data BRI

No	Kata	Jumlah	No	Kata	Jumlah
1	bri	155	11	halo	56
2	saldo	97	12	potong	55
3	atm	87	13	bank	50
4	gagal	74	14	internet	50
5	mohon	74	15	uang	50
6	transaksi	73	16	via	50
7	banking	71	17	aplikasi	49
8	terima	69	18	bantu	44
9	bayar	65	19	mobile	43
10	gagal	74	20	internet	50

Berdasarkan pada Tabel 4.5 dapat diketahui bahwa kata yang paling sering disebut dalam *tweet* akun @kontakBRI adalah kata BRI dengan frekuensi kemunculan sebanyak 155 kali, selanjutnya ada kata saldo yang paling sering disebut dengan frekuensi sebanyak 97 kali. Kata atm merupakan kata ketiga yang memiliki frekuensi paling banyak yakni sebanyak 87 kali serta gagal dan mohon adalah kata terbanyak keempat yang disebut sebanyak 74 kali dalam *tweet* tentang pelayanan Bank BRI.

4.3.2 Klasifikasi Data BRI dengan Regresi Logistik Biner

Variabel Y merupakan variabel klasifikasi *tweet* yang telah dikategorikan dan telah dijelaskan pada Tabel 3.1. Variabel prediktor yang digunakan adalah berupa bobot kata kunci tiap *tweet* yang telah disajikan pada Tabel 4.4. Setelah itu, dilakukan pembagian data *training* dan *testing* dengan menggunakan metode *cross validation* 4 *fold*, data *tweet* sebanyak 459 dibagi menjadi 75% data *training* dan 25% data *testing* sebanyak 4 macam kombinasi (*subset fold*). Selanjutnya akan dilakukan pengujian

independensi terelbih dahulu untuk mengetahui ada tidaknya hubungan antara masing-masing variabel. Hipotesis untuk pengujian korelasi adalah sebagai berikut.

H_0 : tidak terdapat hubungan antara dua variabel

H_1 : terdapat hubungan antara dua variabel

Selanjutnya, berikut adalah hasil pengujian antara variabel respon yakni klasifikasi dengan semua variabel prediktor.

Tabel 4.6. Uji Independensi Variabel Klasifikasi Sentimen dengan Variabel Kata Kunci Data Awal BRI

Klasifikasi dengan	$\hat{\rho}$	**SE($\hat{\rho}$)**	***r***
account	0,014422	0,046712	0,095326
admin	-0,04419	0,046631	0,898169
akses	0,014034	0,046712	0,090266
aktif	0,020213	0,046703	0,187324
⋮	⋮	⋮	⋮
virtual	0,017506	0,046707	0,140469
web	0,016214	0,046709	0,120492
weekend	0,012333	0,046715	0,069701

Berdasarkan pada Tabel 4.6 dapat diketahui bahwa semua variabel predictor tidak memiliki hubungan yang signifikan dengan variabel respon yakni klasifikasi karena nilai *r* yang kurang dari nilai $\chi^2_{(0.05,350)}$ sebesar 394,625. Hal ini dikarenakan karena data yang digunakan tidak seimbang antara sentiment positif dan sentiment negatif sehingga mempengaruhi hasil korelasi. Selanjutnya akan dilakukan uji independensi antar variabel predictor atau antar variabel kata kunci dengan hasil sebagai berikut.

Tabel 4.7. Uji Independensi Antar Variabel Kata Kunci BRI Data Awal

Variabel	***r***	**P*value***
accountxadmin	-0,0199	0,670726
accountxakses	-0,00824	0,860189
accountxaktif	-0,01187	0,79974
⋮	⋮	⋮
weekendxversi	-0,00798	0,864626
weekendxvia	-0,02482	0,595831

Tabel 4.7. Uji Independensi Antar Variabel Kata Kunci BRI Data Awal (Lanjutan)

Variabel	*r*	P*value*
weekendxvirtual	-0,00879	0,850978
weekendxweb	-0,00814	0,86186

Tabel 4.7 menunjukkan bahwa tidak terdapat antar variabel kata kunci yang berhubungan karena nilai P *value* yang kurang dari taraf signifikan sebesar 0,05 sehingga keputusan yang dapat diambil adalah gagal tolak H_0. Hal tersebut juga menunjukkan bahwa tidak adanya kasus multikolinieritas pada data pelayanan BRI.

Selanjutnya akan dilakukan pengujian serentak terlebih dahulu untuk mengetahui data pelayanan nasabah BRI memiliki variabel yang signifikan atau tidak. Data yang digunakan sebagai contoh pengujian merupakan data yang memiliki nilai performa klasifikasi paling tinggi dan hasil dari uji signifikansi parameter pada data pelayanan nasabah BRI yakni data karena pada data SMOTE jumlah data dengan sentimen positif dan negatif sudah di *oversampling* sehingga data menjadi seimbang. Hipotesis yang digunakan dalam pengujian serentak adalah sebagai berikut.

H_0: $\beta_1 = \beta_2 = \ldots = \beta_j = 0$ (tidak ada pengaruh antara variabel kata kunci terhadap variabel klasifikasi sentimen)

H_1: minimal ada satu $\beta_j \neq 0$ (terdapat pengaruh antara variabel kata kunci terhadap variabel klasifikasi sentimen)

Berdasarkan hasil yang diperoleh dapat diketahui bahwa nilai G^2 adalah sebesar 570,7 yang berarti dapat diputuskan tolak H_0 karena nilai G^2 lebih dari nilai $\chi^2_{(0.05,8)}$ sebesar 15,507 sehingga dapat disimpulkan bahwa terdapat pengaruh yang signifikan antara variabel kata kunci terhadap variabel klasifikasi sentimen.

Berikut akan dilakukan uji signifikansi parameter pada data pelayanan nasabah BRI dan variabel prediktornya dengan data yang digunakan sebagai contoh pengujian merupakan data yang memiliki nilai performa klasifikasi paling tinggi dan hasil dari uji signifikansi parameter pada data pelayanan nasabah BRI yakni

data SMOTE dengan *threshold* sebesar 0,5 karena pada data SMOTE jumlah data dengan sentimen positif dan negatif sudah di *oversampling* sehingga data menjadi seimbang. Hipotesis yang digunakan adalah sebagai berikut.

H_0: $\beta_j = 0$ (tidak terdapat pengaruh antara masing-masing variabel kata kunci terhadap variabel respon)

H_1: $\beta_j \neq 0$ (terdapat pengaruh antara masing-masing variabel prediktor terhadap variabel respon)

Berikut adalah hasil dari uji signifikansi parameter yang disajikan pada Tabel 4.8 sebagai berikut.

Tabel 4.8. Uji Signifikansi Parameter BRI

Variabel	***Estimate***	***Std. Error***	***z value***	**P*value***
intercept	1,79783	0,6474	2,777	0,00549
bantu	-3,8959	0,80285	-4,853	$1{,}22 \times 10^{-16}$
call	-6,45085	2,547	-2,533	0,01132
kredit	-5,23124	2,32431	-2,251	0,02441
⋮		⋮	⋮	⋮
ribet	-4,31562	2,14344	-2,013	0,04407
sesuai	-4,97781	1,04883	-4,746	$2{,}07 \times 10^{-6}$
sukses	-4,24437	0,86511	-4,906	$9{,}29 \times 10^{-7}$

Berdasarkan pada hasil di Tabel 4.8 dapat diketahui dengan menggunakan hipotesis nol adalah tidak terdapat pengaruh antara masing-masing kata kunci dengan klasifikasi sentimen, didapatkan keputusan tolak H_0 untuk variabel bantu, *call*, kredit hingga sukses karena nilai P *value* yang kurang dari taraf signifikansi sebesar 0,05 sehingga kesimpulan yang dapat diambil bahwa variabel bantu, *call*, kredit hingga sukses signfikan terhadap model. Selanjutnya dari hasil estimasi parameter koefisien pada Tabel 4.8 dapat ditulis model regresi logistik biner sebagai berikut.

$$\pi(x) = \frac{\exp(1,79 - 3,89X_{12} - 6,45X_{28} + \ldots - 4,24X_{120})}{1 + \exp(1,79 - 3,89X_{12} - 6,45X_{28} + \ldots - 4,24X_{120})}.$$

Sehingga berdasarkan ada model regresi logistik biner tersebut dapat dihitung performa klasifikasi untuk data *training*

dan *testing* pada data pelayanan nasabah BRI. Berikut adalah performa klasifikasi yang didapatkan dengan Regresi Logistik Biner untuk data *tweet* pada pelayanan BRI.

Tabel 4.9. Performa Klasifikasi Regresi Logistik Biner pada Data Awal BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,976	0,00	0,00	0,50
	2	0,976	0,00	0,00	0,50
	3	0,976	0,00	0,00	0,50
	4	0,973	0,00	0,00	0,50
	Rata-Rata	**0,975**	**0,00**	**0,00**	**0,50**
Data *Testing*	1	0,973	0,00	0,00	0,50
	2	0,973	0,00	0,00	0,50
	3	0,973	0,00	0,00	0,50
	4	**0,982**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,975**	**0,00**	**0,00**	**0,50**

Data yang digunakan pada Tabel 4.9 merupakan data awal. Berdasarkan pada Tabel 4.9 tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang sama. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-4 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 98,2% dan nilai AUC sebesar 50%. Sehingga, dapat diketahui bahwa *subset* ke-4 pada data awal adalah *subset* yang memiliki nilai performa klasifikasi terbaik.

Tabel 4.10. *Confusion Matrix* untuk Regresi Logistik Biner pada Data *Testing* Awal BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	0	2
Negatif	0	112

Tabel 4.10 menunjukkan bahwa *confusion matrix* digunakan untuk menghitung ketepatan klasifikasi. Nilai yang berada pada diagonal utama menunjukkan hasil keputusan yang benar sehingga dapat diketahui bahwa ketepatan klasifikasi untuk data negatif yang diklasifikasikan benar negatif adalah 112 data, sedangkan

tidak ada data positif yang diklasifikasikan positif. Terdapat 2 data *tweet* positif yang salah diklasifikasikan ke dalam *tweet* negatif. Selanjutnya akan dihitung performa klasifikasi pada data SMOTE sebagai berikut.

Tabel 4.11. Performa Klasifikasi Regresi Logistik Biner pada Data SMOTE BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,994	0,99	1,00	0,994
	2	0,988	0,98	1,00	0,988
	3	0,991	0,98	1,00	0,991
	4	0,994	0,99	1,00	0,994
	Rata-Rata	**0,992**	**0,985**	**1,00**	**0,992**
Data *Testing*	1	0,982	0,97	1,00	0,982
	2	**0,995**	**0,99**	**1,00**	**0,995**
	3	0,986	0,97	1,00	0,986
	4	0,982	0,97	1,00	0,982
	Rata-Rata	**0,986**	**0,975**	**1,00**	**0,986**

Berdasarkan pada Tabel 4.11 tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang berbeda. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-2 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 99,5%, nilai presisi sebesar 99%, nilai *recall* sebesar 100%, dan nilai AUC sebesar 99,5%. Sehingga, dapat diketahui bahwa *subset* ke-2 pada data SMOTE adalah *subset* terbaik.

Tabel 4.12. *Confusion Matrix* Subset ke-2 untuk Regresi Logistik Biner BRI pada Data *Testing* SMOTE

Kelas Aktual	Kelas Prediksi	
	Positif	Negatif
Positif	113	0
Negatif	1	111

Tabel 4.12 menunjukkan bahwa *confusion matrix* digunakan untuk menghitung ketepatan klasifikasi. Nilai yang berada pada diagonal utama menunjukkan hasil keputusan yang benar sehingga

dapat diketahui bahwa ketepatan klasifikasi untuk data negatif yang diklasifikasikan benar negatif adalah 111 data, sedangkan ada 113 data positif diklasifikasikan positif. Terdapat 1 data *tweet* negatif yang salah diklasifikasikan ke dalam *tweet* positif dan tidak ada data *tweet* positif diklasifikasikan negatif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data *lexicon.*

Tabel 4.13. Performa Klasifikasi Regresi Logistik Biner Bpada Data *Lexicon* RI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,837	0,00	0,00	0,50
	2	0,837	0,00	0,00	0,50
	3	0,837	0,00	0,00	0,50
	4	0,834	0,00	0,00	0,50
	Rata-Rata	**0,836**	**0,00**	**0,00**	**0,50**
Data *Testing*	1	0,834	0,00	0,00	0,50
	2	0,834	0,00	0,00	0,50
	3	0,834	0,00	0,00	0,50
	4	**0,842**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,836**	**0,00**	**0,00**	**0,50**

Berdasarkan pada Tabel 4.13 tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang sama. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-1 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 84,2% dan nilai AUC sebesar 50%. Sehingga, dapat diketahui bahwa *subset* ke-4 pada data *lexicon* adalah *subset* yang memiliki nilai performa klasifikasi terbaik. Berikut adalah *confusion matrix* untuk *subset* ke-4.

Tabel 4.14. *Confusion Matrix* Subset ke-4 untuk Regresi Logistik Biner pada Data *Testing Lexicon* BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	0	16
Negatif	0	99

Tabel 4.14 menunjukkan bahwa data negatif yang diklasifikasikan benar negatif adalah 99 data, sedangkan untuk data

positif yang salah diklasifikasikan negatif adalah sebanyak 16 data *tweet*. Tidak terdapat data *tweet* negatif yang salah diklasifikasikan ke dalam *tweet* positif dan tidak ada data *tweet* positif yang benar diklasifikasikan positif.

4.3.3 Klasifikasi data BRI dengan *Naïve Bayes Classifier*

Metode klasifikasi *Naïve Bayes Classifier* digunakan untuk dapat mengetahui kebaikan dari hasil klasifikasi. Variabel respon atau variabel Y yang digunakan merupakan variabel klasifikasi *tweet* pada BRI yang telah dikategorikan dan telah dijelaskan pada Tabel 3.1. Variabel prediktor yang digunakan adalah berupa bobot kata kunci tiap *tweet* yang telah disajikan pada Tabel 4.2. Berdasarkan metode pembagian data *training* dan *testing* digunakan metode *cross validation* 4 *fold*, data *tweet* yang digunakan adalah sebanyak 459 dibagi menjadi 75% data training dan 25% data testing sebanyak 4 macam kombinasi (*subset fold*). Untuk melihat proses *Naïve Bayes Classifier* pada data *tweet* pelayanan BRI dengan lebih jelas maka akan disajikan ilustrasi perhitungan secara manual dengan menggunakan sampel data yang berasal dari 10 data dengan contoh 350 kata kunci. Untuk pertama kali, dalam melakukan klasifikasi pada data layanan nasabah di BRI akan dicari probabilitas setiap kelas sentimen dengan persamaan (2.24) sebagai berikut.

$$P(V_1) = \frac{|doc_1|}{|training|} = \frac{1}{8} = 0{,}125,$$

$$P(V_2) = \frac{|doc_2|}{|training|} = \frac{1}{8} = 0{,}125,$$

$P(V_1)$ adalah probabilitas sentimen positif dan $P(V_2)$ adalah probabilitas sentimen negatif. Selanjutnya akan dilakukan perhitungan probabilitas kemunculan setiap kata pada masing-masing kategori dengan persamaan (2.25) dengan hasil sebagai berikut.

$$P(x_1 \mid V_1) = \frac{n_1 + 1}{|n + \text{kosakata}|} = \frac{0+1}{|20+350|} = 0,002703,$$

$$P(x_1 \mid V_2) = \frac{n_1 + 1}{|n + \text{kosakata}|} = \frac{0+1}{|52+350|} = 0,002488,$$

$$\vdots$$

$$P(x_{350} \mid V_1) = \frac{n_{350} + 1}{|n + \text{kosakata}|} = \frac{0+1}{|20+350|} = 0,002703,$$

$$P(x_{350} \mid V_2) = \frac{n_{350} + 1}{|n + \text{kosakata}|} = \frac{0+1}{|52+350|} = 0,002488.$$

Selanjutnya, berdasarkan probabilitas kemunculan setiap kata pada masing-masing kategori akan dicai nilai probabilitas tertinggi dari *tweet* yang diujikan. Berikut adalah contoh perhitungan untuk mencari nilai probabilitas tertinggi pada data pelayanan nasabah BRI berdasarkan pada persamaan (2.23).

$$\begin{aligned} P(V_1)\prod_{k=1}^{459} P\left(x_1 \mid \mathrm{V}_j\right) &= (0,125)\times(0,00270\times 0,01081\times 0,00540\times 0,0027 \\ &\quad \times 0,00540\times 0,00270\times 0,00270\times 0,00270\times 0,00270) \\ &= 1,53891\times 10^{-23} \end{aligned}$$

$$\begin{aligned} P(V_2)\prod_{k=1}^{459} P\left(x_1 \mid \mathrm{V}_2\right) &= (0,125)\times(0,00248\times 0,00497\times 0,00248\times 0,00248 \\ &\quad \times 0,00248\times 0,00248\times 0,00248\times 0,00248\times 0,00248) \\ &= 3,84729\times 10^{-24} \end{aligned}$$

Selanjutnya dengan mengulangi persamaan (2.23) tersebut nantinya akan didapatkan nilai V_{MAP} yang merupakan nilai *output* hasil klasifikasi *Naïve Bayes* dari persamaan (2.23) yang paling maksimum untuk setiap *tweet*nya sehingga akan didapatkan ketepatan klasifikasi atau performa klasifikasi dengan menggunakan metode *Naïve Bayes Classifier*. Berikut adalah performa klasifikasi yang didapatkan dengan *Naïve Bayes Classifier* pada data *tweet* pelayanan di BRI berdasarkan pada hasil

klasifikasi dengan perhitungan nilai probabilitas tertinggi untuk setiap *tweet* dengan menggunakan data awal.

Tabel 4.15. Performa Klasifikasi NBC pada Data Awal BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,976	0,00	0,00	0,50
	2	0,973	0,00	0,00	0,498
	3	0,973	0,00	0,00	0,498
	4	0,965	0,00	0,00	0,495
	Rata-Rata	**0,971**	**0,00**	**0,00**	**0,497**
Data *Testing*	1	0,956	0,00	0,00	0,491
	2	**0,973**	**0,00**	**0,00**	**0,50**
	3	0,965	0,00	0,00	0,495
	4	0,973	0,00	0,00	0,495
	Rata-Rata	**0,966**	**0,00**	**0,00**	**0,495**

Tabel 4.15 menunjukkan hasil nilai *accuracy, precision, recall,* dan AUC dengan metode *Naïve Bayes Classifier* pada data awal klasifikasi *tweet* untuk pelayanan Bank BRI. Data yang digunakan merupakan data awal. Berdasarkan pada Tabel 4.15 tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang sama. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-4 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 97,3% dan nilai AUC sebesar 50%. Sehingga, dapat diketahui bahwa *subset* ke-2 pada data awal adalah *subset* yang memiliki nilai performa klasifikasi terbaik. Berikut adalah *confusion matrix* untuk *subset* ke-2.

Tabel 4.16. *Confusion Matrix* Subset ke-2 untuk NBC pada Data *Testing* Awal BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	0	3
Negatif	0	112

Tabel 4.16 menunjukkan bahwa ketepatan klasifikasi untuk data negatif yang diklasifikasikan benar negatif adalah 112 data,

sedangkan tidak ada data positif yang diklasifikasikan positif. Terdapat 3 data *tweet* positif yang salah diklasifikasikan ke dalam *tweet* negatif. Selanjutnya akan dihitung performa klasifikasi pada data SMOTE sebagai berikut.

Tabel 4.17. Performa Klasifikasi NBC pada Data SMOTE BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,988	0,99	0,99	0,988
	2	0,986	0,99	0,99	0,986
	3	0,988	0,99	0,99	0,988
	4	0,988	0,98	1,00	0,988
	Rata-Rata	**0,988**	**0,988**	**0,993**	**0,988**
Data *Testing*	1	**0,991**	**0,99**	**0,99**	**0,991**
	2	0,986	0,97	1,00	0,986
	3	0,991	0,98	1,00	0,991
	4	0,982	0,99	0,97	0,982
	Rata-Rata	**0,988**	**0,983**	**0,990**	**0,988**

Berdasarkan pada Tabel 4.17 tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang berbesa. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-1 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 99,1%, nilai presisi sebesar 99%, nilai *recall* sebesar 99%, dan nilai AUC sebesar 99,1%. Sehingga, *subset* ke-1 pada data SMOTE adalah *subset* yang memiliki nilai performa klasifikasi terbaik.

Tabel 4.18. *Confusion Matrix* Subset ke-1 untuk NBC pada Data *Testing* SMOTE BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	112	1
Negatif	1	111

Tabel 4.18 menunjukkan bahwa *confusion matrix* digunakan untuk menghitung ketepatan klasifikasi. Nilai yang berada pada diagonal utama menunjukkan hasil keputusan yang benar sehingga dapat diketahui bahwa ketepatan klasifikasi untuk data negatif

yang diklasifikasikan benar negatif adalah 111 data, sedangkan terdapat 112 data positif diklasifikasikan positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data *lexicon.*

Tabel 4.19. Performa Klasifikasi NBC pada Data *Lexicon* BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,877	0,82	0,32	0,653
	2	0,860	0,64	0,32	0,643
	3	0,877	0,77	0,36	0,668
	4	0,846	0,57	0,30	0,626
	Rata-Rata	**0,865**	**0,70**	**0,325**	**0,647**
Data *Testing*	1	**0,834**	**0,50**	**0,16**	**0,563**
	2	0,817	0,25	0,05	0,510
	3	0,791	0,27	0,16	0,537
	4	0,833	0,40	0,11	0,539
	Rata-Rata	**0,818**	**0,355**	**0,12**	**0,537**

Tabel 4.19 menunjukkan bahwa hasil nilai akurasi, presisi, *recall* dan AUC dengan metode NBC pada data klasifikasi *tweet* dengan *lexicon* pada pelayanan Bank BRI. Untuk data *testing*, data yang memiliki hasil paling baik adalah data pada *subset* ke-1 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 83,4%, nilai presisi sebesar 82%, nilai *recall* sebesar 32%, dan nilai AUC sebesar 65,3%. Sehingga berdasarkan pada Tabel 4.19 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-1. Berikut adalah *confusion matrix* untuk *subset* ke-1.

Tabel 4.20. *Confusion Matrix* Subset ke-1 untuk NBC pada Data *Testing Lexicon* BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	3	16
Negatif	3	93

Tabel 4.20 menunjukkan hasil bahwa *confusion matrix* digunakan untuk menghitung ketepatan klasifikasi. Nilai yang berada pada diagonal utama merupakan nilai untuk hasil keputusan yang benar. Untuk data negatif yang diklasifikasikan benar negatif

adalah sebanyak 93 data *tweet*, sedangkan untuk data positif yang diklasifikasikan negatif adalah sebanyak 3 data *tweet*. Pada hasil klasifikasi pelayanan Bank BRI dengan menggunakan metode NBC dengan data *lexicon* terdapat sebanyak 16 data *tweet* positif yang diklasifikasikan benar positif dan ada 3 data *tweet* negatif yang diklasifikasikan positif.

4.3.4 Klasifikasi data BRI dengan *Support Vector Machine* Kernel Linier

Metode klasifikasi *Support Vector Machine* (SVM) merupakan salah satu metode yang banyak digunakan untuk analisis klasifikasi. Metode ini sering digunakan karena ketepatan klasifikasi yang dihasilkan cenderung tinggi. Metode pembagian data *training-testing* yang dilakukan dengan menggunakan *kfold cross validation.* Dalam penelitian ini digunakan 4-*fold cross validation* yaitu membagi data keseluruhan menjadi 4 bagian dimana masing-masing bagian akan menjadi *training* dan *testing.* Analisis klasifikasi menggunakan metode SVM dengan kernel linier membutuhkan parameter *cost* (C) yang nantinya akan dilakukan pemilihan nilai parameter yang optimum. Nilai parameter C akan dilakukan uji coba mulai dari 0,01 sampai dengan 100. Data yang digunakan pada metode SVM dengan kernel linier ini digunakan tiga jenis data yakni data awal yang meruakan data dengan klasifikasi variabel respon secara manual, data klasifikasi ppelayanan BRI yang telah dilakukan *oversampling* SMOTE, dan data klasifikasi pelayanan BRI dengan menggunakan *lexicon.* Variabel respon atau variabel Y yang digunakan merupakan variabel klasifikasi *tweet* yang telah dikategorikan dan telah dijelaskan pada Tabel 3.1. Variabel prediktor yang digunakan adalah berupa bobot kata kunci tiap *tweet* yang telah disajikan pada Tabel 4.4. Langkah awal dalam metode SVM dengan kernel linier ini adalah dengan melakukan klasifikasi dengan nilai parameter C yang berbeda-beda. Setelah didapat performa klasifikasi maka selanjutnya akan dipilih nilai C yang paling optimum dengan melihat performa klasifikasi setiap

parameter dan berdasarkan nilai parameter yang optimum akan dipilih performa klasifikasi pada *fold subset* terbaik. Berikut adalah hasil dari performa klasifikasi dengan C yang paling optimum yakni sebesar 0,01.

Tabel 4.21. Performa Klasifikasi SVM Linier pada Data Awal BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,976	0,00	0,00	0,50
	2	0,976	0,00	0,00	0,50
	3	0,976	0,00	0,00	0,50
	4	0,973	0,00	0,00	0,50
	Rata-Rata	**0,975**	**0,00**	**0,00**	**0,50**
Data *Testing*	1	0,973	0,00	0,00	0,50
	2	0,973	0,00	0,00	0,50
	3	0,973	0,00	0,00	0,50
	4	**0,982**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,975**	**0,00**	**0,00**	**0,50**

Tabel 4.21 menunjukkan hasil untuk data *testing* bahwa data awal yang memiliki hasil paling baik adalah data *testing* pada *subset* ke-4 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 98,2% dan nilai AUC sebesar 50%. Sehingga berdasarkan pada Tabel 4.21 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-4. Berikut adalah *confusion matrix* untuk *subset* ke-4.

Tabel 4.22. *Confusion Matrix* Subset ke-4 untuk SVM Linier pada Data *Testing* Awal BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	0	2
Negatif	0	112

Tabel 4.22 menunjukkan hasil bahwa *confusion matrix* digunakan untuk menghitung ketepatan klasifikasi. Untuk data negatif yang diklasifikasikan benar negatif adalah sebanyak 112 data *tweet*, sedangkan untuk data positif yang diklasifikasikan negatif adalah sebanyak 2 data *tweet*. Pada hasil klasifikasi pelayanan Bank BRI dengan menggunakan metode SVM Linier

dengan data awal tidak terdapat data *tweet* positif yang diklasifikasikan benar positif dan tidak ada data *tweet* negatif yang salah diklasifikasikan positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data SMOTE dengan nilai C adalah 0,1 sebagai berikut.

Tabel 4.23. Performa Klasifikasi SVM Linier pada Data SMOTE BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,982	0,97	1,00	0,982
	2	0,982	0,97	1,00	0,982
	3	0,985	0,97	1,00	0,985
	4	0,980	0,96	1,00	0,980
	Rata-Rata	**0,982**	**0,968**	**1,00**	**0,982**
Data *Testing*	1	**0,982**	**0,97**	**1,00**	**0,982**
	2	0,973	0,95	1,00	0,973
	3	0,982	0,97	1,00	0,982
	4	0,982	0,97	1,00	0,982
	Rata-Rata	**0,980**	**0,965**	**1,00**	**0,980**

Tabel 4.23 menunjukkan hasil nilai *accuracy*, *precision*, *recall*, dan AUC pada metode SVM kernel linier dengan nilai parameter C yang optimum yakni 0,1. *Fold subset* terbaik untuk data SMOTE dengan nalai C sebesar 0,01 yakni terdapat pada *subset* ke-4 yang memiliki nilai performa klasifikasi yang terbaik pula yaitu dengan nilai *accuracy* sebesar 98,2%, nilai *precision* sebesar 97%, nilai *recall* sebesar 100%, dan AUC sebesar 98,2%. Selanjutnya dari model SVM kernel linier dengan menggunakan C yang paling optimum yakni sebesar 0,1 dapat dibuat *confusion matrix*. Berikut adalah *confusion matrix* untuk *subset* ke-1 pada data SMOTE.

Tabel 4.24. *Confusion Matrix* untuk SVM Kernel Linear pada Data *Testing* SMOTE BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	113	0
Negatif	4	108

Tabel 4.24 menunjukkan bahwa nilai yang berada di diagonal utama adalah hasil keputusan yang benar sehingga dapat diketahui bahwa ketepatan klasifikasi data negatif yang diklasifikasikan benar negatif adalah sebanyak 108 data *tweet*, sedangkan untuk data positif yang benar diklasifikasikan positif adalah sebanyak 113 data *tweet*. Pada hasil klasifikasi pelayanan Bank BRI ini tidak terdapat data *tweet* positif yang diklasifikasikan negatif dan terdapat 4 data *tweet* negatif yang diklasifikasikan positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data *Lexicon* dengan C yang paling optimum sebesar 0,01 sebagai berikut.

Tabel 4.25. Performa Klasifikasi SVM Linier pada Data *Lexicon* BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,837	0,00	0,00	0,50
	2	0,837	0,00	0,00	0,50
	3	0,837	0,00	0,00	0,50
	4	0,834	0,00	0,00	0,50
	Rata-Rata	**0,836**	**0,00**	**0,00**	**0,50**
Data *Testing*	1	0,834	0,00	0,00	0,50
	2	0,834	0,00	0,00	0,50
	3	0,834	0,00	0,00	0,50
	4	**0,842**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,836**	**0,00**	**0,00**	**0,50**

Tabel 4.35 menunjukkan hasil untuk data *testing* bahwa data *lexicon* yang memiliki hasil paling baik adalah data *testing* pada *subset* ke-1 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 4,2% dan nilai AUC sebesar 50%. Sehingga berdasarkan pada Tabel 4.25 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-4 sebagi berikut.

Tabel 4.26. *Confusion Matrix* untuk SVM Kernel Linear pada Data *Testing Lexicon* BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	0	18
Negatif	0	96

Tabel 4.26 menunjukkan bahwa ketepatan klasifikasi data negatif yang diklasifikasikan benar negatif adalah sebanyak 96 data *tweet*, sedangkan untuk data positif yang benar diklasifikasikan positif tidak ada. Pada hasil klasifikasi pelayanan Bank BRI ini tidak terdapat data *tweet* negatif yang diklasifikasikan positif dan terdapat 18 data *tweet* positif yang diklasifikasikan salah negatif.

Berdasarkan dari keseluruhan jenis data yang digunakan tersebut model yang didapatkan untuk metode SVM kernel linier dengan nilai C sebesar 0,1 yang paling optimum dengan data SMOTE dituliskan pada persamaan berikut.

$$K(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}^T\ \mathrm{x}) + 0{,}1$$

4.3.5 Klasifikasi data BRI *Support Vector Machine* Kernel RBF

Analisis klasifikasi menggunakan metode SVM dengan kernel RBF membutuhkan parameter *cost* (C) dan gamma (γ) yang nantinya akan dilakukan pemilihan nilai parameter yang optimum. Untuk penentuan parameter kernel (γ) dan parameter *cost* (C) digunakan metode *trial and error*. Pada penelitian ini nilai parameter C yang digunakan adalah mulai 0,01 hingga 100 dan parameter gamma (γ) yang akan dilakukan tuning parameter mulai dari 0,01 sampai dengan 100 pada setiap nilai parameter. Pada klasifikasi dengan metode SVM kernel RBF ini diujikan pada tiga jenis data yakni data awal, data dengan *oversampling* SMOTE, dan data dengan klasifikasi *lexicon*. Variabel respon atau variabel Y yang digunakan merupakan variabel klasifikasi *tweet* yang telah dikategorikan dan telah dijelaskan pada Tabel 3.1. Variabel prediktor yang digunakan adalah berupa bobot kata kunci tiap *tweet* yang telah disajikan pada Tabel 4.4. Berdasarkan metode pembagian data *training* dan *testing* digunakan metode *cross validation* 4 *fold*, data *tweet* yang digunakan adalah sebanyak 459 dibagi menjadi 75% data training dan 25% data testing sebanyak 4 macam kombinasi (*subset fold*). Setelah dilakukan klasifikasi

berdasarkan hasil tuning parameter maka nilai C yang paling optimum adalah sebesar 100 dengan nilai gamma (γ) sebesar 0,01 dengan hasil adalah sebagai berikut.

Tabel 4.27. Performa Klasifikasi SVM RBF pada Data Awal BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,985	1,00	0,38	0,687
	2	0,988	1,00	0,50	0,75
	3	0,982	1,00	0,25	0,625
	4	0,976	1,00	0,11	0,555
	Rata-Rata	**0,982**	**1,00**	**0,31**	**0,654**
Data *Testing*	1	0,965	0,00	0,00	0,495
	2	0,973	0,00	0,00	0,50
	3	0,973	0,00	0,00	0,50
	4	**0,982**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,973**	**0,00**	**0,00**	**0,498**

Tabel 4.27 menunjukkan nilai akurasi, presisi, *recall*, dan nilai AUC pada data awal dengan nilai parameter C sebesar 100 dan parameter γ sebesar 0,01. Berdasarkan pada Tabel 4.27 hasil yang diperoleh untuk data *testing* bahwa data awal yang memiliki hasil paling baik adalah data *testing* pada *subset* ke-4 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 98,2% dan nilai AUC sebesar 50%. Sehingga berdasarkan pada Tabel 4.27 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-4. Berikut adalah *confusion matrix* untuk *subset* ke-4.

Tabel 4.28. *Confusion Matrix* Subset ke-4 untuk SVM RBF pada Data *Testing* Awal BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	0	2
Negatif	0	112

Tabel 4.28 menunjukkan hasil *confusion matrix* subset ke-4 untuk SVM kernel RBF BRI pada data *testing* awal. Untuk data negatif yang diklasifikasikan benar negatif adalah sebanyak 112

data *tweet*, sedangkan untuk data positif yang diklasifikasikan negatif adalah sebanyak 2 data *tweet*. Pada hasil klasifikasi pelayanan Bank BRI dengan menggunakan metode SVM kernel RBF dengan data awal tidak terdapat data *tweet* positif yang diklasifikasikan benar positif dan tidak ada data *tweet* negatif yang salah diklasifikasikan positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data SMOTE dengan nilai C adalah 100 dan γ sebesar 0,01 sebagai berikut.

Tabel 4.29. Performa Klasifikasi SVM RBF pada Data SMOTE BRI

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,998	1,00	1,00	0,998
	2	0,997	0,99	1,00	0,997
	3	0,995	0,99	1,00	0,995
	4	0,997	0,99	1,00	0,997
	Rata-Rata	**0,997**	**0,993**	**1,00**	**0,997**
Data *Testing*	1	0,991	0,98	1,00	0,991
	2	0,995	0,99	1,00	0,995
	3	**1,00**	**1,00**	**1,00**	**1,00**
	4	0,991	0,98	1,00	0,991
	Rata-Rata	**0,994**	**0,988**	**1,00**	**0,994**

Tabel 4.29 menunjukkan nilai akurasi, presisi, *recall*, dan nilai AUC pada data *training* dan *testing* pada data SMOTE pelayanan BRI dengan nilai parameter C yang paling optimum adalah sebesar 100 dan parameter γ sebesar 0,01. Berdasarkan pada Tabel 4.29 hasil yang diperoleh untuk data *testing* bahwa data SMOTE yang memiliki hasil paling baik adalah data *testing* pada *subset* ke-3 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 100%, nilai presisi sebesar 100%, nilai *recall* sebesar 100%, dan nilai AUC sebesar 100%. Sehingga berdasarkan pada Tabel 4.29 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-3. Berikut adalah *confusion matrix* untuk *subset* ke-3.

Tabel 4.30. *Confusion Matrix* Subset ke-3 untuk SVM RBF pada Data *Testing* SMOTE BRI

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	113	0
Negatif	0	112

Tabel 4.30 menunjukkan hasil *confusion matrix* subset ke-3 untuk SVM kernel RBF BRI pada data *testing* SMOTE. Untuk data negatif yang diklasifikasikan benar negatif adalah sebanyak 112 data *tweet*, sedangkan terdapat 113 data positif yang benar diklasifikasikan positif. Pada hasil klasifikasi pelayanan Bank BRI dengan menggunakan metode SVM kernel RBF dengan data SMOTE tidak ada data *tweet* positif yang salah diklasifikasikan negatif dan tidak terdapat data negatif diklasifikasikan ke positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data *Lexicon* dengan nilai C adalah 100 dan γ sebesar 0,01 sebagai berikut.

Tabel 4.31. Performa Klasifikasi SVM RBF pada Data *Lexicon* BRI

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,921	0,94	0,55	0,773
	2	0,901	0,92	0,43	0,710
	3	0,918	1,00	0,50	0,75
	4	0,895	0,92	0,40	0,698
	Rata-Rata	**0,908**	**0,94**	**0,47**	**0,732**
Data *Testing*	1	**0,852**	**0,75**	**0,16**	**0,573**
	2	0,826	0,40	0,11	0,537
	3	0,808	0,29	0,11	0,526
	4	0,833	0,33	0,06	0,517
	Rata-Rata	**0,829**	**0,442**	**0,11**	**0,538**

Tabel 4.31 menunjukkan nilai akurasi, presisi, *recall*, dan nilai AUC pada data awal dengan nilai parameter C sebesar 100 dan parameter γ sebesar 0,01. Berdasarkan pada Tabel 4.31 hasil yang diperoleh untuk data *testing* bahwa data *lexicon* yang memiliki hasil paling baik adalah data *testing* pada *subset* ke-1 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar

85,6%, nilai presisi sebesar 75%, nilai *recall* sebesar 16%, dan nilai AUC sebesar 57,3%. Sehingga berdasarkan pada Tabel 4.31 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-1. Berikut adalah *confusion matrix* untuk *subset* ke-1.

Tabel 4.32. *Confusion Matrix* Subset ke-1 untuk SVM RBF pada Data *Testing Lexicon* BRI

Kelas Aktual	Kelas Prediksi	
	Positif	Negatif
Positif	3	16
Negatif	1	95

Tabel 4.32 menunjukkan hasil *confusion matrix* subset ke-1 untuk SVM kernel RBF BRI pada data *testing lexicon*. Untuk data negatif yang diklasifikasikan benar negatif adalah sebanyak 95 data *tweet*, sedangkan untuk data positif yang diklasifikasikan negatif adalah sebanyak 16 data *tweet*. Pada hasil klasifikasi pelayanan Bank BRI dengan menggunakan metode SVM kernel RBF dengan data *lexicon* terdapat 3 data *tweet* positif yang diklasifikasikan benar positif dan ada 1 data *tweet* negatif yang salah diklasifikasikan positif.

Berdasarkan hasil dari semua jenis data yang digunakan nilai C yang paling optimum adalah sebesar 100 dan nilai γ yang digunakan yakni sebesar 0,01 dengan hasil paling optimum, kemudian nilai γ disubstitusikan pada persamaan kernel RBF yang dibahas pada subbab 2.6.2 sehingga akan didapatkan fungsi kernel RBF untuk data pelayanan BRI yang dapat dituliskan pada persamaan berikut.

$$K(\mathrm{x}_1, \mathrm{x}_2) = \exp(-0,01\|\mathrm{x}_1, \mathrm{x}_2\|^2)$$

4.4 Klasifikasi pada data Bank Mandiri

Pada klasifikasi dengan data BRI ini akan dilakukan beberapa metode klasifikasi yang digunakan untuk selanjutnya metode tersebut akan dibandingkan dan dipilih model dengan performa klasifikasi terbaik. Namun sebelumnya data *tweet*

pelayanan BRI setelah dilakukan praproses akan dihitung nilai TF-IDF untuk setiap kata kunci yang telah didapatkan.

4.4.1 TF-IDF untuk data Bank Mandiri

Selanjutnya akan disajikan hasil TF-IDF untuk data *tweet* pada pelayanan Bank Mandiri. Berikut adalah hasil perhitungan TF-IDF untuk data *tweet* pada pelayanan Bank Mandiri yang diambil dari akun Twitter Bank Mandiri dengan *username* @mandiricare yang disajikan pada Tabel 4.33.

Tabel 4.33. TF-IDF pada Data Bank Mandiri

Tweet	aplikasi	atm	banking	⋯	kode	komplain	ramai
1	0	0	0	⋯	0,340	0	0
2	0	0	0	⋯	0,421	0	0
3	0	0	0	⋯	0	0,354	0,419
4	0,458	0	0	⋯	0	0	0
5	0	0,307	0	⋯	0	0	0
6	0	0	0	⋯	0	0	0
7	0	0	0,341	⋯	0	0	0
⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮
2581	0	0	0	⋯	0	0	0
2582	0	0	0	⋯	0	0	0
2583	0	0	0	⋯	0	0	0
2584	0	0	0	⋯	0	0	0
2585	0	0,190	0	⋯	0	0	0
2586	0	0,375	0	⋯	0	0	0
2587	0	0	0	⋯	0	0	0

Data yang telah berbentuk TF-IDF seperti yang disajikan pada Tabel 4.33 tersebut akan digunakan untuk analisis lebih lanjut. Data *tweet* Bank Mandiri memiliki jumlah kata kunci sebanyak 476 kata. Berikut ini disajikan frekuensi kemunculan kata tertinggi dari akun Twitter Bank Mandiri.

Tabel 4.34. Frekuensi Kemunculan Kata pada Data Tweet Mandiri

No	Kata	Jumlah	No	Kata	Jumlah
1	mandiri	1322	10	uang	331
2	saldo	849	11	potong	316
3	atm	675	12	top	316

Tabel 4.34. Frekuensi Kemunculan Kata pada Data Tweet Mandiri (Lanjutan)

No	Kata	Jumlah	No	Kata	Jumlah
4	online	628	13	error	307
5	transaksi	466	14	kasih	299
6	gagal	405	15	Pagi	289
7	ovo	404	16	bantu	276
8	mohon	343	17	banking	263
9	terima	338	18	selamat	244

Berdasarkan pada Tabel 4.34 dapat diketahui bahwa kata yang paling sering disebut dalam *tweet* akun @mandiricare adalah kata Mandiri dengan frekuensi kemunculan sebanyak 1322 kali, selanjutnya ada kata saldo yang paling sering disebut dengan frekuensi sebanyak 849 kali. Kata atm merupakan kata ketiga yang memiliki frekuensi paling banyak yakni sebanyak 675 kali serta *online* adalah kata terbanyak keempat yang disebut sebanyak 628 kali dalam *tweet* tentang pelayanan Bank Mandiri.

4.4.2 Klasifikasi untuk Bank Mandiri dengan Regresi Logistik Biner

Selanjutnya akan dilakukan klasifikasi data *tweet* pelayanan pada Bank Mandiri dengan Regresi Logistik Biner. Variabel Y yang digunakan merupakan variabel klasifikasi *tweet* yang telah dikategorikan dan telah dijelaskan pada Tabel 3.1. Variabel prediktor yang digunakan adalah berupa bobot kata kunci tiap *tweet* yang telah disajikan pada Tabel 4.33. Setelah itu, dilakukan pembagian data *training* dan *testing* dengan menggunakan metode *cross validation* 4 *fold*, data *tweet* sebanyak 2587 dibagi menjadi 75% data *training* dan 25% data *testing* sebanyak 4 macam kombinasi (*subset fold*). Namun terlebih dahulu dilakukan uji signifikansi parameter untuk mengetahui variabel yang signfikan terhadap model. Selanjutnya akan dilakukan pengujian independensi terelbih dahulu untuk mengetahui ada tidaknya hubungan antara masing-masing variabel. Hipotesis untuk pengujian korelasi adalah sebagai berikut.

H_0 : tidak terdapat hubungan antara dua variabel

H_1 : terdapat hubungan antara dua variabel

Selanjutnya, berikut adalah hasil pengujian antara variabel respon yakni klasifikasi dengan semua variabel prediktor.

Tabel 4.35. Uji Independensi Variabel Klasifikasi Sentimen dengan Variabel Kata Kunci pada Data Awal Bank Mandiri

Klasifikasi dengan	$\hat{\rho}$	**SE($\hat{\rho}$)**	***r***
account	0.005885	0.019664	0.089558
admin	0.01277	0.019661	0.421858
akses	0.01507	0.01966	0.58757
aktif	0.008359	0.019663	0.180697
⋮	⋮	⋮	⋮
via	0.011102	0.019662	0.318795
virtual	0.006579	0.019664	0.111924
visa	0.004357	0.019664	0.04909

Berdasarkan pada Tabel 4.35 dapat diketahui bahwa semua variabel predictor tidak memiliki hubungan yang signifikan dengan variabel respon yakni klasifikasi karena nilai *r* yang kurang dari nilai $\chi^2_{(0.05,476)}$ sebesar 527,862. Hal ini dikarenakan karena data yang digunakan tidak seimbang sehingga mempengaruhi hasil korelasi. Selanjutnya akan dilakukan uji independensi antar variabel predictor atau antar variabel kata kunci dengan hasil sebagai berikut.

Tabel 4.36. Uji Independensi Antar Variabel Kata Kunci pada Data Awal Bank Mandiri

Variabel	***r***	**P*value***
accountxadmin	0.015966	0.416959
accountxaju	-0.00324	0.869006
accountxakses	-0.01199	0.542278
accountxaktif	-0.00665	0.735382
accountxaktivasi	-0.00628	0.749508
⋮	⋮	⋮
versixvia	-0.0139	0.479731
versixvirtual	-0.00358	0.855443
versixvisa	-0.00237	0.903966
versixweb	-0.00331	0.866262

Tabel 4.36 menunjukkan bahwa tidak terdapat antar variabel kata kunci yang berhubungan karena nilai P *value* yang kurang dari taraf signifikan sebesar 0,05 sehingga keputusan yang data diambil adalah gagal tolak H_0. Hal tersebut juga menunjukkan bahwa tidak adanya kasus multikolinieritas pada data pelayanan Bank Mandiri.

Selanjutnya akan dilakukan pengujian serentak terlebih dahulu untuk mengetahui data pelayanan nasabah Bank Mandiri memiliki variabel yang signifikan atau tidak. Hipotesis yang digunakan dalam pengujian serentak adalah sebagai berikut.

H_0: $\beta_1 = \beta_2 = \ldots = \beta_j = 0$ (tidak ada pengaruh antara variabel kata kunci terhadap variabel klasifikasi sentimen)

H_1: minimal ada satu $\beta_j \neq 0$ (terdapat pengaruh antara variabel kata kunci terhadap variabel klasifikasi sentimen)

Berdasarkan hasil yang diperoleh dapat diketahui bahwa nilai G^2 adalah sebesar 717,81 yang berarti dapat diputuskan tolak H_0 karena nilai G^2 lebih dari nilai $\chi^2_{(0.05,340)}$ sebesar 527,862 sehingga dapat disimpulkan bahwa terdapat pengaruh yang signifikan antara variabel kata kunci terhadap variabel klasifikasi sentimen.

Uji signifikansi parameter pada data pelayanan nasabah Bank Mandiri dan variabel prediktornya ini merupakan salah satu langkah awal dalam tahap regresi logistic biner dengan menggunakan data contoh yakni data *lexicon* sebagai berikut.

Tabel 4.37. Uji Signifikansi Parameter pada Data Bank Mandiri

Variabel	***Estimate***	***Std. Error***	***z value***	**P*value***
(Intercept)	$-2{,}95 \times 10^{0}$	$6{,}91 \times 10^{-1}$	-4,266	$1{,}99 \times 10^{-5}$
aju	$1{,}27 \times 10^{1}$	$6{,}01 \times 10^{0}$	2,113	0,03462
besok	$2{,}45 \times 10^{1}$	$1{,}11 \times 10^{1}$	2,196	0,02807
duit	$4{,}69 \times 10^{0}$	$2{,}29 \times 10^{0}$	2,052	0,04017
edc	$9{,}83 \times 10^{0}$	$4{,}51 \times 10^{0}$	2,181	0,02916
⋮	⋮	⋮	⋮	⋮
tunggu	$8{,}27 \times 10^{0}$	$3{,}15 \times 10^{0}$	2,625	0,00866
uang	$3{,}05 \times 10^{0}$	$1{,}12 \times 10^{0}$	2,721	0,00651
user	$1{,}24 \times 10^{1}$	$4{,}89 \times 10^{0}$	2,532	0,01133

Berdasarkan pada hasil di Tabel 4.37 dapat diketahui bahwa dengan menggunakan hipotesis nol adalah tidak terdapat pengaruh antara kata kunci dengan klasifikasi sentimen maka, didapatkan keputusan tolak H_0 untuk variabel besok, duit, *edc*, *emoney*, hingga user sehingga kesimpulan yang dapat diambil bahwa variabel kata kunci dari variabel besok, duit, edc, emoney, hingga user signfikan terhadap model. Selanjutnya hasil estimasi parameter koefisien yang signifikan terhadap model dapat ditulis model regresi logistik biner sebagai berikut.

$$\pi(x) = \frac{\exp(-2,95 \times 10 + 1,27 \times 10 X_{5} + 2,45 \times 10 X_{53} + \ldots + 1,24 \times 10 X_{463})}{1 + \exp(-2,95 \times 10 + 1,27 \times 10 X_{5} + 2,45 \times 10 X_{53} + \ldots + 1,24 \times 10 X_{463})}$$

Sehingga berdasarkan pada langkah-langkah setelah mendapatkan model regresi logistik biner tersebut dapat dihitung performa klasifikasi untuk data *training* dan data *testing* pada data pelayanan nasabah Bank Mandiri. Berikut adalah performa klasifikasi yang didapatkan dengan Regresi Logistik Biner untuk data *tweet* pada pelayanan Bank Mandiri untuk data awal.

Tabel 4.38. Performa Klasifikasi Regresi Logistik Biner pada Data Awal Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,992	0,00	0,00	0,50
	2	0,992	0,00	0,00	0,50
	3	0,992	0,00	0,00	0,50
	4	0,992	0,00	0,00	0,50
	Rata-Rata	**0,992**	**0,00**	**0,00**	**0,50**
Data *Testing*	1	0,992	0,00	0,00	0,50
	2	0,992	0,00	0,00	0,50
	3	0,992	0,00	0,00	0,50
	4	**0,993**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,992**	**0,00**	**0,00**	**0,50**

Tabel 4.38 menunjukkan hasil nilai *accuracy*, *precision*, *recall*, dan AUC dengan metode Regresi Logistik Biner pada data klasifikasi *tweet* untuk pelayanan Bank Mandiri. Data yang digunakan merupakan data awal. Berdasarkan pada Tabel 4.38

tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang sama. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-4 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 99,3% dan nilai AUC sebesar 50%. Sehingga, dapat diketahui bahwa *subset* ke-4 pada data awal adalah *subset* yang memiliki nilai performa klasifikasi terbaik. Berikut adalah *confusion matrix* untuk *subset* ke-4.

Tabel 4.39. *Confusion Matrix* Subset ke-4 untuk Regresi Logistik Biner Bank Mandiri pada Data *Testing* Awal

Kelas Aktual	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	0	4
Negatif	0	642

Tabel 4.39 menunjukkan bahwa ketepatan klasifikasi untuk data negatif yang diklasifikasikan benar negatif adalah 642 data, sedangkan tidak ada data positif yang diklasifikasikan positif. Terdapat 4 data *tweet* positif yang salah diklasifikasikan ke dalam *tweet* negatif dan tidak terdaat *tweet* negatif yang diklasifikasikan salah menjadi *tweet* positif. Selanjutnya akan dihitung performa klasifikasi pada data SMOTE sebagai berikut.

Tabel 4.40. Performa Klasifikasi Regresi Logistik Biner pada Data SMOTE Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,994	0,99	1,00	0,994
	2	0,993	0,99	1,00	0,993
	3	0,994	0,99	1,00	0,994
	4	0,992	0,99	1,00	0,992
	Rata-Rata	**0,993**	**0,990**	**1,00**	**0,993**
Data *Testing*	1	0,989	0,98	1,00	0,989
	2	0,991	0,98	1,00	0,991
	3	0,987	0,98	1,00	0,987
	4	**0,995**	**0,99**	**1,00**	**0,995**
	Rata-Rata	**0,991**	**0,983**	**1,00**	**0,991**

Berdasarkan pada Tabel 4.40 tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang berbeda. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-4 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 99,5%, nilai presisi sebesar 99%, nilai *recall* sebesar 100%, dan nilai AUC sebesar 99,5%. Sehingga, *subset* ke-4 pada data SMOTE adalah *subset* yang memiliki nilai performa klasifikasi terbaik.

Tabel 4.41. *Confusion Matrix* Subset ke-4 untuk Regresi Logistik Biner pada Data *Testing* SMOTE Bank Mandiri

Kelas Aktual	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	641	0
Negatif	6	636

Tabel 4.41 menunjukkan bahwa ketepatan klasifikasi untuk data negatif yang diklasifikasikan benar negatif adalah 636 data, sedangkan untuk data positif diklasifikasikan positif adalah sebanyak 641 data *tweet*. Tidak terdapat data *tweet* positif yang salah diklasifikasikan ke dalam *tweet* negatif dan ada 6 data *tweet* negatif diklasifikasikan positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data *lexicon.*

Tabel 4.42. Performa Klasifikasi Regresi Logistik Biner pada Data *Lexicon* Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,836	0,85	0,28	0,632
	2	0,837	0,85	0,29	0,636
	3	0,831	0,84	0,25	0,620
	4	0,842	0,82	0,33	0,655
	Rata-Rata	**0,836**	**0,84**	**0,287**	**0,635**
Data *Testing*	1	0,797	0,56	0,24	0,594
	2	0,816	0,70	0,23	0,603
	3	**0,828**	**0,82**	**0,24**	**0,613**
	4	0,804	0,82	0,10	0,548
	Rata-Rata	**0,811**	**0,725**	**0,202**	**0,589**

Tabel 4.42 menunjukkan hasil perhitungan performa klasifikasi dilakukan dengan menggunakan data *training* dan data *testing* untuk selanjutnya dihitung nilai rata-ratanya. Berdasarkan pada Tabel 4.42 tersebut data *testing* awal memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-3 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 82,8%, nilai presisi sebesar 82%, nilai *recall* sebesar 24%, dan nilai AUC sebesar 61,3%. Sehingga, dapat diketahui bahwa *subset* ke-3 pada data awal adalah *subset* yang memiliki nilai performa klasifikasi terbaik. Selanjutnya akan digunakan hasil *confusion matrix* berdasarkan pada *subset* terbaik yakni *subset* 3. Berikut adalah *confusion matrix* untuk *subset* ke-3.

Tabel 4.43. *Confusion Matrix* Subset ke-3 untuk Regresi Logistik Biner pada Data *Testing Lexicon* Bank Mandiri

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	33	104
Negatif	7	502

Tabel 4.43 menunjukkan bahwa *confusion matrix* digunakan untuk menghitung ketepatan klasifikasi. Nilai yang berada pada diagonal utama menunjukkan hasil keputusan yang benar sehingga dapat diketahui bahwa ketepatan klasifikasi untuk data negatif yang diklasifikasikan benar negatif adalah 502 data *tweet*, sedangkan untuk data positif diklasifikasikan positif ada sebanyak 33 data *tweet*. Untuk data positif yang salah diklasifikasikan negatif ada sebanyak 104 data *tweet* dan untuk data *tweet* negatif diklasifikasikan positif terdapat sebanyak 7 data *tweet*.

4.4.3 Klasifikasi untuk Bank Mandiri dengan *Naïve Bayes Classifier*

Selanjutnya akan dilakukan klasifikasi *tweet* pelayanan pada bank Mandiri. Langkah-langkah yang diterapkan sama dengan melakukan klasifikasi data pelayanan nasabah pada BRI yang telah dilakukan sebelumnya mulai dari mencari probabilitas setiap kelas

sentimen lalu menghitung probabilitas kemunculan setiap kata pada masing-masing kategori dan selanjutnya mendapatkan hasil klasifikasi dengan perhitungan nilai probabilitas tertinggi untuk setiap *tweet*. Variabel respon yang digunakan merupakan variabel klasifikasi *tweet* pada Bank Mandiri yang telah dikategorikan dan telah dijelaskan pada Tabel 3.1. Variabel prediktor yang digunakan adalah berupa bobot kata kunci tiap *tweet* yang telah disajikan pada Tabel 4.33. Berdasarkan metode pembagian data *training* dan *testing* digunakan metode *cross validation* 4 *fold*, data *tweet* yang digunakan dibagi menjadi 75% data *training* dan 25% data *testing* sebanyak 4 macam kombinasi (*subset fold*). Berikut adalah performa klasifikasi dengan *Naïve Bayes Classifier* pada data *tweet* pelayanan di Bank Mandiri pada data awal.

Tabel 4.44. Performa Klasifikasi NBC pada Data Awal Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,992	0,00	0,00	0,50
	2	0,992	0,00	0,00	0,50
	3	0,992	0,00	0,00	0,50
	4	0,992	0,00	0,00	0,533
	Rata-Rata	**0,992**	**0,00**	**0,00**	**0,507**
Data *Testing*	1	0,990	0,00	0,00	0,499
	2	0,992	0,00	0,00	0,50
	3	0,992	0,00	0,00	0,50
	4	**0,993**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,992**	**0,00**	**0,00**	**0,50**

Tabel 4.44 menunjukkan hasil nilai *accuracy*, *precision*, *recall*, dan AUC dengan metode *Naïve Bayes Classifier* pada data awal klasifikasi *tweet* untuk pelayanan Bank Mandiri. Data yang digunakan merupakan data awal. Berdasarkan pada Tabel 4.44 tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang sama. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-4 memiliki nilai performa klasifikasi yang terbaik yaitu dengan nilai akurasi sebesar 99,3% dan nilai AUC sebesar 50%. Sehingga, dapat diketahui bahwa *subset* ke-4 pada data awal adalah *subset* yang

memiliki nilai performa klasifikasi terbaik. Berikut adalah *confusion matrix* untuk *subset* ke-4.

Tabel 4.45. *Confusion Matrix* Subset ke-4 untuk NBC pada Data *Testing* Awal Bank Mandiri

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	0	4
Negatif	0	642

Tabel 4.45 menunjukkan bahwa ketepatan klasifikasi untuk data negatif yang diklasifikasikan benar negatif adalah 642 data, sedangkan tidak ada data positif yang diklasifikasikan positif. Terdapat 4 data *tweet* positif yang salah diklasifikasikan ke dalam *tweet* negatif dan tidak terdapat *tweet* negatif yang salah diklasifikasikan ke *tweet* positif. Selanjutnya akan dihitung performa klasifikasi pada data SMOTE sebagai berikut.

Tabel 4.46. Performa Klasifikasi NBC pada Data SMOTE Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,996	0,99	1,00	0,996
	2	0,995	0,99	1,00	0,995
	3	0,997	1,00	1,00	0,997
	4	0,996	0,99	1,00	0,996
	Rata-Rata	**0,996**	**0,993**	**1,00**	**0,996**
Data *Testing*	1	0,995	0,99	1,00	0,995
	2	**0,997**	**1,00**	**1,00**	**0,997**
	3	0,994	0,99	1,00	0,994
	4	0,996	0,99	1,00	0,996
	Rata-Rata	**0,996**	**0,993**	**1,00**	**0,996**

Berdasarkan pada Tabel 4.46 tersebut data *training* dan *testing* memiliki rata-rata performa klasifikasi yang berbeda. Untuk data *testing* memiliki rata-rata performa klasifikasi paling tinggi yakni pada *subset* ke-2 memiliki nilai performa klasifikasi dengan nilai akurasi sebesar 99,7%, nilai presisi sebesar 100%, nilai *recall* sebesar 100%, dan nilai AUC sebesar 99.7%. Sehingga, *subset* ke-2 pada data SMOTE adalah *subset* terbaik.

Tabel 4.47. *Confusion Matrix* Subset ke-2 untuk NBC pada Data *Testing* SMOTE Bank Mandiri

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	641	0
Negatif	3	639

Tabel 4.47 menunjukkan bahwa nilai yang berada pada diagonal utama menunjukkan hasil keputusan yang benar sehingga dapat diketahui bahwa ketepatan klasifikasi untuk data negatif yang diklasifikasikan benar negatif adalah 639 data, sedangkan terdapat 641 data positif diklasifikasikan positif. Tidak terdapat data *tweet* positif yang salah diklasifikasikan ke dalam *tweet* negative dan terdapat 3 data negatif yang diklasifikasikan ke positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data *lexicon.*

Tabel 4.48. Performa Klasifikasi NBC pada Data *Lexicon* Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,824	0,62	0,45	0,685
	2	0,819	0,60	0,46	0,689
	3	0,819	0,60	0,45	0,686
	4	0,819	0,59	0,51	0,706
	Rata-Rata	**0,820**	**0,602**	**0,467**	**0,691**
Data *Testing*	1	0,754	0,43	0,45	0,643
	2	0,755	0,42	0,39	0,620
	3	**0,818**	**0,61**	**0,41**	**0,669**
	4	0,775	0,44	0,20	0,564
	Rata-Rata	**0,775**	**0,475**	**0,362**	**0,624**

Tabel 4.48 menunjukkan bahwa hasil nilai *accuracy* dan AUC dengan metode NBC pada data klasifikasi *tweet* pada pelayanan Bank Mandiri. Dimana data *lexicon* yang memiliki hasil paling baik adalah dengan *fold* pada *subset* ke-3 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 81,8%, nilai presisi sebesar 61%, nilai *recall* sebesar 41%, dan nilai AUC sebesar 66,9%. Berikut adalah *confusion matrix* untuk *subset* ke-3.

Tabel 4.49. *Confusion Matrix* Subset ke-3 untuk NBC Bank Mandiri pada Data *Testing Lexicon*

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	56	81
Negatif	36	473

Tabel 4.49 menunjukkan hasil bahwa untuk data negatif yang diklasifikasikan benar negatif adalah sebanyak 473 data *tweet*, sedangkan untuk data positif diklasifikasikan benar positif ada sebanyak 56 data *tweet*. Untuk data positif yang salah diklasifikasikan negatif ada sebanyak 81 data *tweet* dan untuk data *tweet* negatif yang salah diklasifikasikan positif terdapat sebanyak 36 data *tweet*.

4.4.4 Klasifikasi data Bank Mandiri dengan *Support Vector Machine* Kernel Linier

Selanjutnya akan dilakukan klasifikasi pada pelayanan nasabah Bank Mandiri dengan metode SVM kernel linier dengan nilai parameter yang diujikan yakni mulai dari C sebesar 0,01 hingga 100. Berikut adalah hasil performa klasifikasi dengan C sebesar 0,01 yang merupakan parameter paling optimum sebagai berikut.

Tabel 4.50. Performa Klasifikasi SVM Linier pada Data Awal Bank Mandiri

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,992	0,00	0,00	0,50
	2	0,992	0,00	0,00	0,50
	3	0,992	0,00	0,00	0,50
	4	0,992	0,00	0,00	0,50
	Rata-Rata	**0,992**	**0,00**	**0,00**	**0,50**
Data *Testing*	1	0,992	0,00	0,00	0,50
	2	0,992	0,00	0,00	0,50
	3	0,992	0,00	0,00	0,50
	4	**0,993**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,992**	**0,00**	**0,00**	**0,50**

Tabel 4.50 menunjukkan hasil untuk data *testing* bahwa data awal yang memiliki hasil paling baik adalah data *testing* dengan C sebesar 0,01 pada *subset* ke-4 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 99,2% dan nilai AUC sebesar 50%. Sehingga berdasarkan pada Tabel 4.50 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-4. Berikut adalah *confusion matrix* untuk *subset* ke-4.

Tabel 4.51. *Confusion Matrix* Subset ke-4 untuk SVM Linier pada Data *Testing* Awal Bank Mandiri

Kelas Aktual	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	0	4
Negatif	0	642

Tabel 4.51 menunjukkan hasil bahwa *confusion matrix* untuk data negatif yang diklasifikasikan benar negatif adalah sebanyak 642 data *tweet*, sedangkan untuk data positif yang diklasifikasikan negatif adalah sebanyak 4 data *tweet*. Pada hasil klasifikasi pelayanan Bank Mandiri ini dengan menggunakan metode SVM Linier dengan data awal tidak terdapat data *tweet* positif yang diklasifikasikan benar positif dan tidak ada data *tweet* negatif yang salah diklasifikasikan positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data SMOTE dengan nilai C adalah 0,01.

Tabel 4.52. Performa Klasifikasi SVM Linier pada Data SMOTE Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,935	0,98	0,89	0,935
	2	0,937	0,98	0,90	0,937
	3	0,936	0,98	0,90	0,936
	4	0,937	0,98	0,90	0,937
	Rata-Rata	**0,936**	**0,980**	**0,898**	**0,936**
Data *Testing*	1	0,933	0,98	0,88	0,933
	2	0,929	0,97	0,88	0,929
	3	**0,943**	**0,98**	**0,91**	**0,943**
	4	0,938	0,98	0,90	0,938
	Rata-Rata	**0,936**	**0,978**	**0,893**	**0,936**

Tabel 4.52 menunjukkan hasil nilai *accuracy, precision, recall,* dan AUC dengan metode SVM kernel linier dengan nilai parameter C yang optimum yakni 0,01 pada data pelayanan Bank Mandiri yang telah di SMOTE. *Fold subset* terbaik untuk data SMOTE dengan nalai C sebesar 0,01 yakni terdapat pada *subset* ke-3 yang memiliki nilai performa klasifikasi yang terbaik pula yaitu dengan nilai *accuracy* sebesar 94,3%, nilai *precision* sebesar 98%, nilai *recall* sebesar 91%, dan AUC sebesar 94,3%. Selanjutnya dari model SVM kernel linier dengan menggunakan C yang paling optimum yakni sebesar 0,01 dapat dibuat *confusion matrix* untuk melihat ketepatan klasifikasi pada data pelayan di Bank Mandiri.

Tabel 4.53. *Confusion Matrix* untuk SVM Kernel Linear pada Data *Testing* SMOTE Bank Mandiri

Kelas Aktual	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	582	59
Negatif	14	628

Tabel 4.53 menunjukkan hasil bahwa *confusion matrix*. Terdapat data negatif yang diklasifikasikan benar negatif sebanyak 628 data *tweet*, sedangkan untuk data positif yang benar diklasifikasikan positif ada sebanyak 582 data *tweet*. Untuk data positif yang salah diklasifikasikan negatif ada sebanyak 59 data *tweet* dan untuk data *tweet* negatif yang diklasifikasikan positif ada sebanyak 14 data *tweet*. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data *Lexicon* dengan C yang paling optimum sebesar 0,1.

Tabel 4.54. Performa Klasifikasi SVM Linier pada Data *Lexicon* Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,793	1,00	0,02	0,512
	2	0,789	1,00	0,01	0,504
	3	0,787	0,00	0,00	0,50
	4	0,793	0,93	0,03	0,515
	Rata-Rata	**0,790**	**0,732**	**0,015**	**0,507**

Tabel 4.54. Performa Klasifikasi SVM Linier pada Data *Lexicon* Bank Mandiri (Lanjutan)

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Testing*	1	**0,791**	**0,80**	**0,03**	**0,513**
	2	0,789	1,00	0,01	0,503
	3	0,787	0,00	0,00	0,50
	4	0,787	0,00	0,00	0,50
	Rata-Rata	**0,788**	**0,45**	**0,01**	**0,504**

Tabel 4.54 menunjukkan hasil untuk data *testing* bahwa data *lexicon* yang memiliki hasil paling baik adalah data *testing* pada *subset* ke-1 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 79,1% dan nilai AUC sebesar 51,3%. Sehingga berdasarkan pada Tabel 4.54 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-1. Berikut adalah *confusion matrix* untuk *subset* ke-1.

Tabel 4.55. *Confusion Matrix* untuk SVM Kernel Linear pada Data *Testing Lexicon* Bank Mandiri

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	4	134
Negatif	1	509

Tabel 4.55 menunjukkan bahwa ketepatan klasifikasi data negatif yang diklasifikasikan benar negatif adalah sebanyak 509 data *tweet*, sedangkan untuk data positif yang benar diklasifikasikan positif ada 4 data. Pada hasil klasifikasi pelayanan Bank Mandiri ini terdapat 1 data *tweet* negatif yang salah diklasifikasikan positif serta terdapat 134 data *tweet* positif yang salah diklasifikasikan pada *tweet* negatif.

Untuk pembetukan model SVM dengan kernel linier ada data pelayanan Bank Mandiri berdasarkan dari keseluruhan jenis data yang digunakan, model yang didapatkan untuk metode SVM kernel linier dengan nilai C sebesar 0,01 yang paling optimum dituliskan pada persamaan berikut.

$$K(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}^T\ \mathrm{x}) + 0,01$$

4.4.5 Klasifikasi data Bank Mandiri dengan *Support Vector Machine* Kernel RBF

Selanjutnya akan dilakukan klasifikasi data *tweet* pelayanan pada Bank Mandiri dengan *Support Vector Machine* dengan kernel RBF. Variabel Y yang digunakan merupakan variabel klasifikasi *tweet* yang telah dikategorikan dan telah dijelaskan pada Tabel 3.1. Variabel prediktor yang digunakan adalah berupa bobot kata kunci tiap *tweet* yang telah disajikan pada Tabel 4.33. Setelah itu, dilakukan pembagian data *training* dan *testing* dengan menggunakan metode *cross validation* 4 *fold*, data *tweet* sebanyak 2587 dibagi menjadi 75% data *training* dan 25% data *testing* sebanyak 4 macam kombinasi (*subset fold*). Berikut adalah performa klasifikasi data awal dengan *Support Vector Machine* dengan kernel RBF untuk data *tweet* pada pelayanan Bank Mandiri setelah dilakukan klasifikasi berdasarkan tuning parameter dan didapatkan nilai C yang paling optimum adalah sebesar 100 dan nilai gamma yang diujikan ternyata menghasilkan nilai paling optimum pada gamma sebesar 0,01 sebagai berikut.

Tabel 4.56. Performa Klasifikasi SVM RBF pada Data Awal Bank Mandiri

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Training*	1	0,993	1,00	0,07	0,535
	2	0,993	1,00	0,14	0,571
	3	0,993	1,00	0,14	0,571
	4	0,994	1,00	0,33	0,666
	Rata-Rata	**0,993**	**1,00**	**0,17**	**0,585**
Data *Testing*	1	0,992	0,00	0,00	0,50
	2	0,992	0,00	0,00	0,50
	3	0,992	0,00	0,00	0,50
	4	**0,993**	**0,00**	**0,00**	**0,50**
	Rata-Rata	**0,992**	**0,00**	**0,00**	**0,50**

Tabel 4.56 menunjukkan nilai akurasi, presisi, *recall*, dan nilai AUC pada data awal dengan nilai parameter C sebesar 100 dan parameter γ sebesar 0,01. Berdasarkan pada Tabel 4.56 hasil yang diperoleh untuk data *testing* bahwa data awal yang memiliki hasil paling baik adalah data *testing* pada *subset* ke-4 yang

memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 99,3% dan nilai AUC sebesar 50%. Sehingga berdasarkan pada Tabel 4.56 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-4. Berikut adalah *confusion matrix* untuk *subset* ke-4.

Tabel 4.57. *Confusion Matrix* Subset ke-4 untuk SVM RBF pada Data *Testing* Awal Bank Mandiri

Kelas Aktual	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	0	4
Negatif	0	642

Tabel 4.57 menunjukkan hasil *confusion matrix subset* ke-4 untuk SVM kernel RBF Bank Mandiri pada data *testing* awal. Untuk data negatif yang diklasifikasikan benar negatif adalah sebanyak 642 data *tweet*, sedangkan untuk data positif yang diklasifikasikan salah negatif adalah sebanyak 4 data *tweet*. Pada klasifikasi pelayanan Bank Mandiri dengan metode SVM kernel RBF dengan data awal tidak terdapat data *tweet* positif yang diklasifikasikan benar positif dan tidak ada data *tweet* negatif yang salah diklasifikasikan positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data SMOTE dengan nilai C adalah 10 dan γ sebesar 0,01.

Tabel 4.58. Performa Klasifikasi SVM RBF pada Data SMOTE Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,992	0,99	1,00	0,992
	2	0,992	0,98	1,00	0,992
	3	0,991	0,98	1,00	0,991
	4	0,994	0,99	1,00	0,994
	Rata-Rata	**0,992**	**0,985**	**1,00**	**0,992**
Data *Testing*	1	0,992	0,99	1,00	0,992
	2	0,991	0,98	1,00	0,991
	3	**0,996**	**0,99**	**1,00**	**0,996**
	4	0,983	0,97	1,00	0,983
	Rata-Rata	**0,991**	**0,983**	**1,00**	**0,991**

Tabel 4.58 menunjukkan nilai akurasi, presisi, *recall*, dan nilai AUC pada data *training* dan *testing* SMOTE dengan nilai parameter C sebesar 10 dan parameter γ sebesar 0,01. Berdasarkan pada Tabel 4.58 hasil yang diperoleh untuk data *testing*, data SMOTE yang memiliki hasil paling baik adalah data *testing* pada *subset* ke-3 yang memiliki nilai performa klasifikasi yaitu nilai akurasi sebesar 99,6%, nilai presisi sebesar 99%, nilai *recall* sebesar 100%, dan nilai AUC sebesar 99,6%. Sehingga berdasarkan pada Tabel 4.58 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-3. Berikut adalah *confusion matrix* untuk *subset* ke-3.

Tabel 4.59. *Confusion Matrix* Subset ke-3 untuk SVM RBF pada Data *Testing* SMOTE Bank Mandiri

Kelas Aktual	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	641	0
Negatif	5	637

Tabel 4.59 menunjukkan hasil *confusion matrix* subset ke-3 untuk SVM kernel RBF Bank Mandiri pada data *testing* SMOTE. Data negatif yang diklasifikasikan benar negatif adalah sebanyak 637 data, sedangkan untuk data positif yang benar diklasifikasikan positif adalah sebanyak 641 data juga. Tidak terdapat data *tweet* positif yang salah diklasifikasikan negatif dan ada 5 data *tweet* negatif yang salah diklasifikasikan pada data *tweet* positif. Selanjutnya akan dihitung performa klasifikasi dengan menggunakan data *Lexicon* dengan nilai C adalah 10 dan γ sebesar 0,01 sebagai berikut.

Tabel 4.60. Performa Klasifikasi SVM RBF pada Data *Lexicon* Bank Mandiri

Data	***Fold***	**Akurasi**	**Presisi**	***Recall***	***AUC***
Data *Training*	1	0,813	0,92	0,13	0,564
	2	0,812	0,89	0,14	0,565
	3	0,814	0,88	0,14	0,568
	4	0,816	0,90	0,15	0,574
	Rata-Rata	**0,814**	**0,898**	**0,14**	**0,568**

Tabel 4.60. Performa Klasifikasi SVM RBF pada Data *Lexicon* Bank Mandiri (Lanjutan)

Data	*Fold*	Akurasi	Presisi	*Recall*	*AUC*
Data *Testing*	1	0,804	0,82	0,10	0,547
	2	**0,816**	**0,91**	**0,15**	**0,571**
	3	0,808	0,88	0,11	0,552
	4	0,814	0,90	0,14	0,567
	Rata-Rata	**0,811**	**0,878**	**0,125**	**0,559**

Tabel 4.60 menunjukkan nilai akurasi, presisi, *recall*, dan nilai AUC pada data awal dengan nilai parameter C sebesar 10 dan parameter γ sebesar 0,01. Berdasarkan pada Tabel 4.58 *fold subset* terbaik yakni pada *subset* ke-2 yang memiliki nilai performa klasifikasi sebesar 81,6% untuk nilai akurasi, 91% untuk nilai *precision*, 15% untuk nilai *recall* dan 57,1% untuk nilai AUC. Sehingga berdasarkan pada Tabel 4.60 tersebut, *confusion matrix* akan disusun berdasarkan *fold* pada *subset* ke-2.

Tabel 4.61. *Confusion Matrix* Subset ke-2 untuk SVM RBF pada Data *Testing Lexicon* Bank Mandiri

Kelas Aktual	Kelas Prediksi	
	Positif	**Negatif**
Positif	20	356
Negatif	2	508

Tabel 4.61 menunjukkan hasil bahwa *confusion matrix* digunakan untuk menghitung ketepatan klasifikasi. Untuk data negatif yang diklasifikasikan benar negatif adalah sebanyak 508 data *tweet*, sedangkan untuk data positif diklasifikasikan benar positif ada sebanyak 20 data *tweet*. Terdapat 356 data *tweet* positif yang salah diklasifikasikan negatif dan terdapat 2 data *tweet* negatif yang salah diklasifikasikan positif.

Nilai C yang paling optimum adalah sebesar 10 dan nilai γ yang digunakan yakni sebesar 0,01 akan digunakan untuk menentukan model, sehingga fungsi kernel RBF untuk data pelayanan Bank Mandiri dapat dituliskan pada persamaan berikut.

$$K(\mathrm{x}_1, \mathrm{x}_2) = \exp(-0,01\|\mathrm{x}_1, \mathrm{x}_2\|^2)$$

4.5 Pemilihan Metode Terbaik

Untuk mengetahui metode yang terbaik digunakan pada klasifikasi *tweet* pelayanan Bank BRI maka selanjutnya akan dilakukan perbandingan nilai performa klasifikasi untuk memperoleh metode klasifikasi yang terbaik. Berdasarkan hasil pengulangan klasifikasi BRI sebanyak 3 kali maka didapatkan hasil rata-rata performa klasifikasi (akurasi, presisi, dan AUC) yang ditunjukkan pada Tabel 4.62.

Tabel 4.62. Pemilihan Metode Terbaik pada Data Pelayanan BRI

Data	Metode	Jenis	Akurasi	Presisi	*AUC*
Data Awal	RLB	*Training*	0,975	0,00	0,50
		Testing	0,975	0,00	**0,50**
	NBC	*Training*	0,971	0,00	0,498
		Testing	0,966	0,00	0,495
	SVM Linier	*Training*	0,975	0,00	0,50
		Testing	0,975	0,00	**0,50**
	SVM RBF	*Training*	0,982	1,00	0,654
		Testing	0,973	0,00	0,498
Data SMOTE	RLB	*Training*	0,991	0,985	0,991
		Testing	0,985	0,973	0,985
	NBC	*Training*	0,987	0,987	0,987
		Testing	0,987	0,981	0,987
	SVM Linier	*Training*	0,983	0,970	0,983
		Testing	0,978	0,960	0,977
	SVM RBF	*Training*	0,997	0,993	0,995
		Testing	0,992	0,984	**0,992**
Data dengan *Lexicon*	RLB	*Training*	0,836	0,00	0,50
		Testing	0,836	0,00	0,50
	NBC	*Training*	0,865	0,70	0,647
		Testing	0,818	0,355	**0,537**
	SVM Linier	*Training*	0,836	0,00	0,50
		Testing	0,836	0,00	0,50
	SVM RBF	*Training*	0,908	0,94	0,732
		Testing	0,829	0,442	0,538

Berdasarkan pada Tabel 4.62 tersebut terlihat bahwa pada metode SVM dengan kernel linier dan RBL memiliki nilai AUC paling tinggi pada data *testing* data awal dengan rata-rata AUC sebesar 50% untuk kedua metode. Selanjutnya yang memiliki nilai paling tinggi untuk melakukan klasifikasi pada pelayanan nasabah Bank BRI terdapat pada metode SVM dengan kernel RBF jika menggunakan data SMOTE dengan rata-rata AUC sebesar 99,2% untuk data *testing*. Untuk data *lexicon* AUC paling tinggi untuk data *training* terdapat pada metode SVM kernel RBF dan untuk data *testing* terdapat pada metode NBC dengan rata-rata AUC data *testing* sebesar 53,7%. Secara keseluruhan, metode yang paling cocok digunakan untuk melakukan klasifikasi data pelayanan nasabah BRI adalah dengan metode SVM kernel RBF karena nilai AUC yang dimiliki selalu paling tinggi dengan menggunakan data SMOTE. Karena data awal dan data *lexicon* memiliki proporsi yang tidak seimbang sehingga data SMOTE lebih baik digunakan. Selanjutnya, akan dibandingkan performa klasifikasi data pelayanan di Bank Mandiri.

Tabel 4.63. Pemilihan Metode Terbaik pada Data Pelayanan Bank Mandiri

Data	Metode	Jenis	Akurasi	Presisi	*AUC*
Data Awal	RLB	*Training*	0,992	0,00	0,50
		Testing	0,992	0,00	**0,50**
	NBC	*Training*	0,992	0,00	0,507
		Testing	0,992	0,00	**0,50**
	SVM Linier	*Training*	0,992	0,00	0,50
		Testing	0,992	0,00	**0,50**
	SVM RBF	*Training*	0,993	1,00	0,585
		Testing	0,992	0,00	**0,50**
Data SMOTE	RLB	*Training*	0,993	0,990	0,993
		Testing	0,991	0,983	0,991
	NBC	*Training*	0,996	0,992	0,996
		Testing	0,996	0,993	**0,996**
	SVM Linier	*Training*	0,937	0,978	0,937
		Testing	0,936	0,976	0,936
	SVM RBF	*Training*	0,992	0,986	0,992
		Testing	0,990	0,983	0,990

Tabel 4.63. Pemilihan Metode Terbaik pada Data Pelayanan Bank Mandiri (Lanjutan)

Data	Metode	Jenis	Akurasi	Presisi	*AUC*
Data dengan *Lexicon*	RLB	*Training*	0,836	0,84	0,635
		Testing	0,811	0,725	0,589
	NBC	*Training*	0,820	0,602	0,691
		Testing	0,775	0,475	**0,624**
	SVM Linier	*Training*	0,790	0,732	0,507
		Testing	0,788	0,45	0,504
	SVM RBF	*Training*	0,814	0,898	0,568
		Testing	0,811	0,878	0,559

Berdasarkan hasil pengulangan klasifikasi sebanyak 3 kali maka didapatkan hasil rata-rata performa klasifikasi (akurasi, presisi, dan AUC) yang ditunjukkan pada Tabel 4.63. Sehingga berdasarkan Tabel 4.63 tersebut dapat diketahui bahwa hasil rata-rata performa klasifikasi untuk setiap metode yang digunakan memiliki hasil yang berbeda-beda. Dengan menggunakan data awal dapat diketahui bahwa semua metode menghasilkan nilai AUC yang sama yakni sebesar 50%. Untuk data SMOTE klasifikasi dengan metode NBC memiliki nilai AUC sempurna untuk data *training* dan untuk data *testing* memiliki nilai AUC yang sangat baik. Sedangkan pada data *lexicon* juga memiliki hasil bahwa metode NBC juga memiliki nilai AUC tertinggi untuk data *training* sebesar 69,1% dan data *testing* sebesar 62,4%. Namun, jika dianalisa lebih dalam terjadi kasus *overfitting* pada hasil data awal dengan data SMOTE. Disini, hasil dari SMOTE menghasilkan performa klasifikasi yang jauh lebih tinggi daripada dengan data awal. Seharusnya performa klasifikasi data SMOTE tidak jauh berbeda dengan data awal. Namun pada penelitan ini, jumlah data pada data awall dan data SMOTE yang mengakibatkan hal tersebut terjadi. Jika pada data awal, perbandingan data yang sangat tidak seimbang membuat nilai AUC tidak begitu baik. Namun, setelah dilakukan SMOTE data sudah memiliki jumlah yang seimbang dan jumlah replikasinya sebesar 40 kali data minornya sehingga menghasilkan performa klasifikasi yang jauh lebih baik. Sehingga secara keseluruhan, metode *Naïve Bayes*

Classifier patut dipilih atau layak digunakan dalam melakukan klasifikasi untuk pelayanan nasabah Bank Mandiri dengan menggunakan data yang telah di SMOTE.

4.6 Visualisasi *Word Cloud* Pada Akun Twitter Setiap Bank

Word cloud disini akan menampilkan kata-kata yang sering muncul dari data *tweet* pelayanan nasabah BRI dan Bank Mandiri sehingga berdasarkan kata-kata yang sering muncul tersebut dapat diketahui saran atau peningkatan yang harus dilakukan oleh masing-masing bank.

4.6.1 *Word Cloud* BRI

Word cloud pada data *tweet* pelayanan nasabah BRI ini akan dibedakan menjadi unigram dan bigram. Unigram merupakan kata kunci yang terambil dari hanya satu suku kata sedangkan bigram merupakan dari dua suku kata.

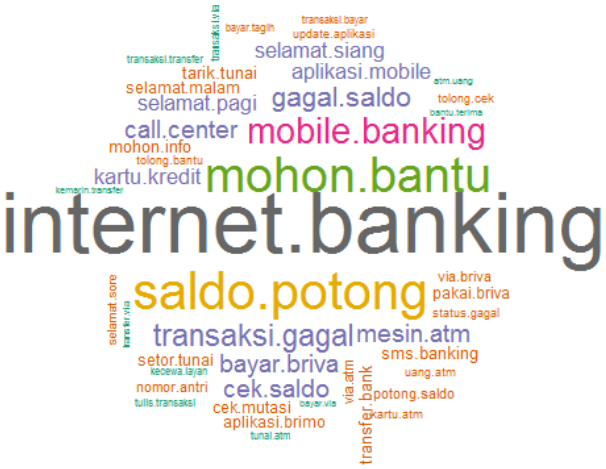
(a) (b)

Gambar 4.2 Word Cloud Unigram Sentimen Positif BRI (a) *Word Cloud* Unigram Sentimen Negatif BRI (b)

Gambar 4.2 (a) pada *word cloud* sentimen positif menunjukan bahwa kata Bank BRI yang diambil dari akun @kontakBRI, yang sering muncul adalah kata “mudah”, “sukses”, dan “respon” dalam *tweet* pelayanan nasabah BRI dengan sentimen

positif. Hal tersebut mengindikasikan bahwa nasabah sering melakukan apresiasi terhadap pelayanan BRI. Namun jika dilihat pada Gambar 4.2 (b) yang merupakan *word cloud* sentimen negatif untuk BRI dapat diketahui bahwa kata "gagal", "potong", dan "error" merupakan kata yang sering muncul dalam *tweet* bersentimen negatif. Hal tersebut berarti bahwa banyak *tweet* nasabah yang mengalami kegagalan dalam bertransaksi.

Mengingat pada Gambar 4.1 lebih banyak *tweet* yang bersentimen negatif daripada positif hal tersebut artinya banyak sekali orang atau nasabah yang melakukan pengaduan kepada akun Twitter BRI terkait kegagalan, terpotongnya saldo, dan *error* saat melakukan transaksi. Berdasarkan Gambar 4.3 dibawah ini kata internet *banking*" merupakan kata yang paling sering disebut dalam *tweet* nasabah BRI. Para nasabah kebanyakan mengeluhkan tentang aplikasi dari internet *banking* yang tidak berjalan dengan lancar. Banyak terjadi kasus saldo sudah terpotong namun saldo tidak masuk ke rekening tujuan atau tidak masuk dalam proses transaksi.

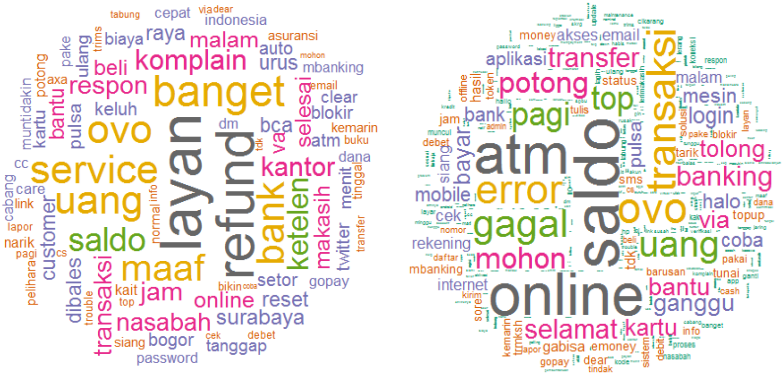


Gambar 4.3 *Word Cloud* Bigram Sentimen Negatif BRI

Berdasarkan pada Gambar 4.3 dapat diketahui pula bahwa kata “internet banking” merupakan kata yang paling banyak muncul. Lalu ada kata “saldo potong”, “transaksi gagal”, “gagal saldo”, dan “mohon bantu” yang juga sering muncul. Hal tersebut berarti bahwa pada Gambar 4.3 tersebut terdapat kata-kata negatif yang mengindikasikan bahwa terdapat banyak *tweet* dengan sentimen negatif didalamnya. Internet banking sering muncul dikarenakan banyak nasabah yang melakukan komplainnya terkait aplikasi bri mobile atau yang biasa disebut internet banking pada saat transaksi. Banyak pula yang mengeluhkan saldo terpotong sendirinya tetapi saldo yang terpotong tidak masuk ke rekening tujuan dikarenakan adanya *error* pada sistem internet banking atau sistem lainnya.

4.6.2 *Word Cloud* Bank Mandiri

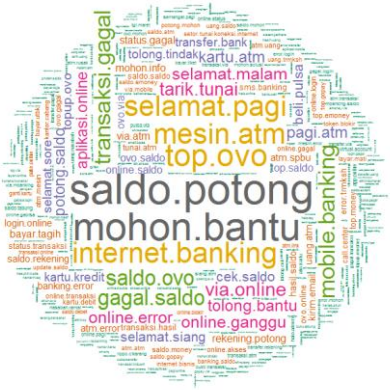
Word cloud pada data *tweet* pelayanan nasabah Mandiri ini akan dibedakan pula antara unigram dan bigram. Dari data yang telah dipraproses sebelumnya, maka visualisasi w*ord cloud* untuk data *tweet* nasabah di Bank Mandiri.



(a) (b)

Gambar 4.4 *Word Cloud* Unigram Sentimen Positif Mandiri (a) *Word Cloud* UnigramSentimen Negatif Mandiri (b)

Gambar 4.4 (a) menunjukan bahwa kata “layan” dan “*refund*” merupakan kata yang sangat dominan atau sering kali muncul dalam *tweet* pelayanan Bank Mandiri yang bersentimen positif. Untuk Gambar 4.4 (b) menunjukkan bahwa kata “*online*” dan “saldo” merupakan kata yang paling banyak muncul dalam data *tweet* pelayanan nasabah Bank Mandiri. Kata tersebut muncul dalam *tweet* dengan sentiment negatif paling banyak. Jika diteliti lebih lanjut banyak nasabah yang melakukan pengaduan atau komplain kepada akun Twitter Bank Mandiri dikarenakan aplikasi mandiri *online* yang terkadang sering melakukan *maintenance* dan berakibat gagalnya proses transaksi. Hal ini didukung dengan kata selanjutnya yang sering muncul adalah “gagal” dan “error” merupakan kata bersentimen negatif. Hal ini dapat mengindikasikan bahwa nasabah sering menggunakan kata tersebut dalam *tweet* untuk melakukan komplain terhadap kegagalan transaksi mandiri *online*. Selanjutnya untuk dapat melihat kata-kata yang yang tersusun dalam dua kata pada data *tweet* pelayanan nasabah Bank Mandiri yang paling sering muncul maka akan disajikan Gambar 4.5 berikut.



Gambar 4.5 *Word Cloud* Bigram Sentimen Negatif Mandiri

Berdasarkan pada Gambar 4.5 tersebut dapat diketahui bahwa kata “saldo potong” serta “*internet banking*” merupakan

kata yang paling sering muncul dalam data *tweet* pelayanan nasabah Bank Mandiri. Hal tersebut sejalan dengan kata yang sering muncul pada hasil Gambar 4.4 yang juga kata "*refund*" dan "saldo" yang paling sering muncul. Pengaduan tentang mandiri *online* ini kebanyakan terjadi karena saldo yang terpotong namun saldo tidak masuk kedalam transaksi yang dituju sehingga banyak nasabah yang melakukan komplainnya. Jika dilihat lebih lanjut pada data *tweet* pelayanan nasabah Bank Mandiri dengan sentimen negatif kebanyakan merupakan komplain tentang adanya masalah pada aplikasi mandiri *online*. Masalah atau adanya gagal transaksi pada mandiri *online* sering terjadi saat pihak Bank Mandiri melakukan perawatan terhadap aplikasi dan berujung merugikan nasabah. Para nasabah yang melakukan complain kebanyakan pada saat hari kerja dan siang hari ketika aplikasi mandiri *online*nya tidak dapat digunakan. Oleh karena itu, sebaiknya jika akan melakukan perawatan terhadap aplikasi mandiri *online* sebaiknya pihak Mandiri melakukan disaat jam tidak aktif atau saat transaksi nasabah sedang minimum guna menghindari komplain yang banyak dari nasabah.

Halaman ini sengaja dikosongkan

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan sebelumnya mengenai klasifikasi data pelayanan nasabah BRI dan Bank Mandiri, maka kesimpulan yang dapat diambil adalah sebagai berikut :

1. Hasil klasifikasi data awal, data SMOTE, dan data *lexicon* memiliki hasil terbaik pada data SMOTE untuk data layanan BRI dan Bank Mandiri.
2. Hasil klasifikasi dengan Regresi Logistik Biner diketahui memiliki akurasi terbaik untuk BRI adalah sebesar 08,5% dengan nilai AUC sebesar 98,5% dengan data SMOTE. Untuk Bank Mandiri nilai akurasinya adalah sebesar 99,3% dengan nilai AUC 99,3% dengan data SMOTE.
3. Hasil klasifikasi menggunakan metode *Naïve Bayes Classifier* mempunyai nilai AUC sebesar 98,7% untuk BRI dengan data SMOTE dan 99,6% untuk Bank Mandiri dengan data SMOTE.
4. Untuk hasil klasifikasi menggunakan SVM diperoleh kernel yang menghasilkan akurasi dan AUC paling tinggi adalah dengan kernel RBF jika menggunakan SMOTE untuk BRI dan Bank Mandiri. Nilai AUC untuk BRI adalah sebesar 99,2% dan untuk Bank Mandiri adalah sebesar 99%.
5. Metode terbaik untuk mengklasifikasikan sentimen layanan pada BRI adalah SMOTE-SVM dengan kernel RBF, sedangkan untuk Bank Mandiri adalah dengan SMOTE-NBC karena memiliki nilai AUC yang paling tinggi.

5.2 Saran

Pada penelitian ini, saran yang dapat diberikan adalah dapat memberikan informasi kepada pihak Bank baik BRI maupun Bank Mandiri untuk selalu meningkatkan sistem layanan pada transaksi berbasis *online* dan *offline*, karena masih banyak terjadi kegagalan

transaksi terutama pada transaksi berbasis *online.* Saran untuk penelitian selanjutnya adalah jika menggunakan klasifikasi berdasarkan pada kamus *lexicon,* perlu diperhatikan daftar kata positif dan daftar kata negatif yang dimasukkan agar hasil klasifikasi dapat memberikan hasil yang tepat dan akurat, karena pada penelitian masih banyak *tweet* yang kurang tepat pengklasifikasiannya dengan menggunakan kamus *lexicon.* Hal lain yang perlu diperhatikan yakni, jika data tidak seimbang maka untuk Regresi Logistik Biner sebaiknya digunakan *threshold* yang sesuai agar hasil klasifikasi dapat sesuai dan menghasilkan performa klasifikasi yang tinggi. Selanjutnya untuk data yang tidak seimbang jika tidak dilakukan SMOTE akan menghasilkan performa klasifikasi yang sangat kecil sehingga sangat dianjurkan untuk melakukan SMOTE pada data yang tidak seimbang.

DAFTAR PUSTAKA

Anwar, S. (2015). *Analisis Sentimen Pada Akun Twitter Provider Telekomunikasi.* Bandung: Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia.

Bekkar, M., Djemaa, K. D., & Alitouche, A. D. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications*.

BRI. (2019, February 6). *PT.Bank Rakyat Indonesia (Persero) Tbk.* Retrieved from Sejarah: https://bri.co.id/sejarah

Brownlee, J. (2019, February 2). Retrieved from How to Prepare Text Data for Machine Learning with Scikit-Learn: https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. (2002). SMOTE : Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, Vol 16, pp. 321-357.

Durajati, C., & Gumelar, A. B. (2012). Pemanfaatan Teknik Supervised untuk Klasifikasi Teks Bahasa Indonesia. Vol. 16, pp. 1-8.

Forbes. (2018). *The World's Largest Public Companies*. Retrieved from Forbes: https://www.forbes.com/global2000/list/

Gokgoz, E., & Subasi, A. (2015). Comparison of Decision Tree Algorithms for EMG Signal Classification using DWT. *Biomedical Signal Processing and Control*, Vol 18, pp. 138-144.

Gorunescu, F. (2011). *Data Mining.* Verlag Berlin Heidelberg: Springer.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques.* USA: Elsevier Inc.

Hardle, W. K., Prastyo, D. D., & Hafner, C. M. (2013). Support Vector Machines with Evolutionary Model Selection for Default Prediction.

Hosmer, D., & Stanley, L. (2000). *Applied Logistic Regression.* New York: John Wiley & Sons, Inc.

Hu, M., & Liu, B. (2004, March 2). *Opinion Mining, Sentiment Analysis, and Opinion Spam Detection*. Retrieved from https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

Jin, Y. (2016). Development of Word Cloud Generator Software Based on Python.

Karimah, P. A. (2018). *Analisis Kompetitif Sosial Media Menggunakan Metode Klasifikasi Naive Bayes Classifier (NBC) dan Support Vector Machine (SVM): Studi Kasus di Industri Transportasi Umum Taksi.* Surabaya: Departemen Statistika, FMKSD, Institut Teknologi Sepuluh Nopember.

Kumar, S. N. (2015). World towards Advance Web Mining: A Review. *American Journal of Systems and Software*, Vol 3, pp. 44-61.

Kurniawan, T. (2013). *Implementasi Text Mining Pada Analisis Media Mainstream Menggunakan Naive Bayes Classifier dan Support Vector Machine.* Surabaya: Departemen Statistika, FMIPA, Institut Teknologi Sepuluh Nopember.

Manalu, B. U. (2014). *Analisis Sentimen Pada Twitter Menggunakan Text Mining.* Medan: Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Mayasari, R. W. (2018). *Text Mining Pada Akun Resmi Pemerintah Kota Surabaya dengan Metode Regresi Logistik Biner, Support Vector Machine (SVM), dan Naive Bayes Classifier (NBC).* Surabaya: Departemen Statistika, FMKSD, Institut Teknologi Sepuluh Nopember.

OJK. (2018). *Booklet Perbankan Indonesia*. Jakarta: Departemen Perizinan dan Informasi Perbankan,Otoritas Jasa Keuangan.

Sain, H., & Purnami, S. W. (2015). Combine Sampling Support Vector Machine for Imbalanced Data Classification. *Procedia Computer Science*, Vol. 72, pp. 59-66.

Tala, F. Z. (2003). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia.* Amsterdam: Institute for Logic, Language and Computation, Universiteit van Amsterdam.

Talib, R., Hanif, K. M., Ayesha, S., & Fatima, F. (2016). Text Mining: Techniques, Applications and Issues. *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 7, pp. 11.

Tan, P. N., Steinbach, M., Karpatne, A., & Kumar, V. (2018). *Introduction to Data Mining (2nd Edition).* United States: Pearson Education.

Halaman ini sengaja dikosongkan

LAMPIRAN

Lampiran 1. Syntax Crawling

```
library(twitteR)
library(rtweet)
library(tidyverse)
library(tm)

create_token(
  app             <- "ernadns",
  consumer_key    <- "HdyBGJR4r74ihrUuIJZjqquDF ",
  consumer_secret <-
"fFEfxu1kLEVjDpv5jmz8aYaMRsARHnHZ62es8GPUl4bvpJ8Vyn ",
  access_token    <- "306680209-
V900XFtwn4jacO8pGrQBT7cxhbbOqrdlE47yZedJ",
  access_secret   <-
"4OdRgoVGM2LDegAShcsVqujXgb22E5ul7sQUxxtzysyBS")

mandiri <- search_tweets("mandiricare", lang="id", n=10000,
tweet_mode="extended", include_rts=FALSE)
crawmandiri <- data.frame(mandiri$created_at, mandiri$screen_name,
mandiri$text, mandiri$display_text_width,
as.character(mandiri$mentions_screen_name))
write.csv(mandiri, file="crawmandiri.csv")
bri <- search_tweets("kontakBRI", lang="id", n=10000,
tweet_mode="extended", include_rts=FALSE)
crawbri <- data.frame(bri$created_at, bri$screen_name, bri$text,
bri$display_text_width, as.character(bri$mentions_screen_name))
write.csv(bri, file="crawbri.csv")
```

Lampiran 2. Syntax Praproses Data

```
import pandas as pd
import string
import nltk
from nltk.tokenize import word_tokenize
import re
import sys
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

#Memasukkan Data
data= pd.read_csv("E:\KULIAH\TUGAS AKHIR LJ\FULL
TWEET\DATA UNTUK BRI TANPA REPLIES\data BRI bersih rev.csv",
engine="python")
pre_tweet=data["Full tweet"]

#Clear Link 1
dataclearlink=[]
for line in pre_tweet:
    result=re.sub(r" .twitter\S+","",line)
    dataclearlink.append(result)
    print(result)
#Clear Link 2
dataclearlink1=[]
for line in dataclearlink:
    result=re.sub(r"pic\S+","",line)
    dataclearlink1.append(result)
    print(result)
#Clear Link 3
dataclearlink2=[]
for line in dataclearlink1:
    result=re.sub(r"http\S+","",line)
    dataclearlink2.append(result)
    print(result)
```

Lampiran 2. Syntax Praproses Data (Lanjutan)

```
#Clear Hastag
dataclearhastag=[]
for line in dataclearlink2:
    result=re.sub(r"#\S+","",line)
    dataclearhastag.append(result)
    print(result)

#Menghapus Simbol Retweet
dataclearrt = []
for line in dataclearhastag:
    result = re.sub(r"RT", " ", line)
    dataclearrt.append(result)
    print(result)

#Menghapus Username
dataclearusername = []
for line in dataclearrt:
    result = re.sub(r"@\S+", " ", line)
    dataclearusername.append(result)
    print(result)

#Menghapus angka
dataclearangka = []
for line in dataclearusername :
    result = re.sub("\d", " ", line)
    dataclearangka.append(result)
    print(result)

#Menghapus emoticon
dataclearemoticon = []
for line in dataclearangka :
    result = re.sub(r'<.*?>', " ", line)
    dataclearemoticon.append(result)
    print(result)
```

Lampiran 2. Syntax Praproses Data (Lanjutan)

```
#Menghapus baris baru
dataclearline = []
for line in dataclearemoticon :
    result = re.sub("\n", " ", line)
    dataclearline.append(result)
    print(result)

#Menghapus punctuation
dataclearpunctuation = []
for line in dataclearline :
    result = re.sub(r"[^\w\s]", " ", line)
    dataclearpunctuation.append(result)
    print(result)

#Menghapus spasi berlebih
datacleardoublespace = []
for line in dataclearpunctuation :
    result=re.sub(r'\s+', ' ', line)
    datacleardoublespace.append(result)
    print(result)

#Menghapus Underscore
dataclearunderscore = []
for line in datacleardoublespace :
    result = re.sub(r"\S+_", " ", line)
    dataclearunderscore.append(result)
    print(result)

#Casefolding
datalower=[]
for line in dataclearunderscore:
    a=line.lower()
    datalower.append(a)
    print(a)
```

Lampiran 2. Syntax Praproses Data (Lanjutan)

```
#Stemming
factory=StemmerFactory()
stemmer=factory.create_stemmer()
datastemmed=map(lambda x: stemmer.stem(x), datalower)
databersih=list(datastemmed)
print(databersih)

#Stopwords
stopwords=open('stopwords.txt', 'r').read()
datafinal=[]
df=[]
for line in databersih:
    wo = word_tokenize(line)
    wo = [word for word in wo if not word in stopwords and not word[0].isdigit()]
    datafinal.append(wo)
    df.append(" ".join(wo))
    print(datafinal)

#CountVectorize
from sklearn.feature_extraction.text import TfidfTransformer
from pandas import DataFrame
from sklearn.feature_extraction.text import CountVectorizer
Y = data['Klasifikasi Awal']
Y_nya = pd.DataFrame(Y)
vect = CountVectorizer(min_df=2, max_df=200)
X = vect.fit_transform(df)
X_=DataFrame(X.A, columns=vect.get_feature_names())
pembobotan = TfidfTransformer(use_idf=True).fit_transform(X)
hasil=DataFrame(pembobotan.A, columns=vect.get_feature_names())

#TFIDF
tfidf = TfidfTransformer(use_idf=True).fit_transform(X_)
tfidf_nya = (tfidf.toarray())
X_nya = tfidf_nya
tf=DataFrame(tfidf.A, columns=vect.get_feature_names())
print (tf)
```

Lampiran 3. Syntax Klasifikasi

```
#Klasifikasi
import numpy
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import optunity
import optunity.metrics
from scipy import stats
import sklearn
from sklearn.metrics import classification_report, confusion_matrix
from __future__ import print_function
from sklearn.model_selection import train_test_split, cross_val_score, StratifiedKFold, KFold, ShuffleSplit, StratifiedShuffleSplit
from sklearn.feature_selection import SelectPercentile, f_classif
from sklearn.naive_bayes import BernoulliNB
from sklearn.svm import LinearSVC, SVC
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, accuracy_score
from sklearn.utils import shuffle
from sklearn import svm, datasets
from sklearn.metrics import roc_curve, auc
from scipy import interp
from imblearn.over_sampling import SMOTE
from sklearn import metrics
import sklearn.svm
```

Lampiran 4. Syntax Klasifikasi Regresi Logistik Biner

```
#Regresi Logistik Biner
kfold= StratifiedKFold(n_splits=4, shuffle=True)
kfold.get_n_splits(X_new)
kfold.get_n_splits(Y_new)
lr= LogisticRegression()
akurasi=[]

for train, test in kfold.split(X_new, Y_new):
        (X_new[train], X_new[test])
        (Y_new[train], Y_new[test])
        hitung = lr.fit(X_new[train], Y_new[train])
        predictrbl1 = lr.predict(X_new[train])
        predictrbl = lr.predict(X_new[test])
        fpr1, tpr1, _1 = metrics.roc_curve(Y_new[train], predictrbl1)
        fpr, tpr, _ = metrics.roc_curve(Y_new[test], predictrbl)
        auc_1 = metrics.auc(fpr1,tpr1)
        auc_ = metrics.auc(fpr,tpr)
        print("\nConfusiion Matrix Train: \n",
metrics.confusion_matrix(Y_new[train], predictrbl1))
        print("\nConfusiion Matrix Test: \n",
metrics.confusion_matrix(Y_new[test], predictrbl))
        akurasi.append(accuracy_score(Y_new[train], predictrbl1))
        akurasi.append(accuracy_score(Y_new[test], predictrbl))
        print("Akurasi Prediksi Train: ", accuracy_score(Y_new[train],
predictrbl1))
        print("Akurasi Prediksi Test: ", accuracy_score(Y_new[test],
predictrbl))
        print()
        print(metrics.classification_report(Y_new[train], predictrbl1))
        print(metrics.classification_report(Y_new[test], predictrbl))
        print("AUC Train: ", auc_1)
        print("AUC Test: ", auc_)
```

Lampiran 5. Syntax Klasifikasi *Naïve Bayes Classifier*

```
#NBC
kfold= StratifiedKFold(n_splits=4, shuffle=True)
kfold.get_n_splits(X_new)
kfold.get_n_splits(Y_new)
cl = BernoulliNB()
akurasi = []

for train, test in kfold.split(X_new, Y_new):
        (X_new[train], X_new[test])
        (Y_new[train], Y_new[test])
        hitung = cl.fit(X_new[train], Y_new[train])
        predictnbc1 = cl.predict(X_new[train])
        predictnbc = cl.predict(X_new[test])
        fpr1, tpr1, _1 = metrics.roc_curve(Y_new[train], predictnbc1)
        fpr, tpr, _ = metrics.roc_curve(Y_new[test], predictnbc)
        auc_1 = metrics.auc(fpr1,tpr1)
        auc_ = metrics.auc(fpr,tpr)
        print("\nConfusiion Matrix Train: \n",
metrics.confusion_matrix(Y_new[train], predictnbc1))
        print("\nConfusiion Matrix Test: \n",
metrics.confusion_matrix(Y_new[test], predictnbc))
        akurasi.append(accuracy_score(Y_new[train], predictnbc1))
        akurasi.append(accuracy_score(Y_new[test], predictnbc))
        print("Akurasi Prediksi Train: ", accuracy_score(Y_new[train],
predictnbc1))
        print("Akurasi Prediksi Test: ", accuracy_score(Y_new[test],
predictnbc))
        print()
        print(metrics.classification_report(Y_new[train], predictnbc1))
        print(metrics.classification_report(Y_new[test], predictnbc))
        print("AUC Train: ", auc_1)
        print("AUC Test: ", auc_)
```

Lampiran 6. Syntax Klasifikasi SVM

```
#SVM Kernel Linear
kfold= StratifiedKFold(n_splits=4, shuffle=True)
kfold.get_n_splits(X_new)
kfold.get_n_splits(Y_new)
lne= SVC(C=0.1, kernel='linear')
akurasi = []

for train, test in kfold.split(X_new, Y_new):
        (X_new[train], X_new[test])
        (Y_new[train], Y_new[test])
        hitung = lne.fit(X_new[train], Y_new[train])
        predictlne1 = lne.predict(X_new[train])
        predictlne = lne.predict(X_new[test])
        fpr1, tpr1, _1 = metrics.roc_curve(Y_new[train], predictlne1)
        fpr, tpr, _ = metrics.roc_curve(Y_new[test], predictlne)
        auc_1 = metrics.auc(fpr1,tpr1)
        auc_ = metrics.auc(fpr,tpr)
        print("\nConfusiion Matrix Train: \n",
metrics.confusion_matrix(Y_new[train], predictlne1))
        print("\nConfusiion Matrix Test: \n",
metrics.confusion_matrix(Y_new[test], predictlne))
        akurasi.append(accuracy_score(Y_new[train], predictlne1))
        akurasi.append(accuracy_score(Y_new[test], predictlne))
        print("Akurasi Prediksi Train: ", accuracy_score(Y_new[train],
predictlne1))
        print("Akurasi Prediksi Test: ", accuracy_score(Y_new[test], predictlne))
        print()
        print(metrics.classification_report(Y_new[train], predictlne1))
        print(metrics.classification_report(Y_new[test], predictlne))
        print("AUC Train: ", auc_1)
        print("AUC Test: ", auc_)
```

Lampiran 6. Syntax Klasifikasi SVM (Lanjutan)

```
#SVM Kernel RBF
kfold= StratifiedKFold(n_splits=4, shuffle=True)
kfold.get_n_splits(X_new)
kfold.get_n_splits(Y_new)
rbf= SVC(C=100, gamma=0.01, kernel='rbf')
akurasi = []
for train, test in kfold.split(X_new, Y_new):
        (X_new[train], X_new[test])
        (Y_new[train], Y_new[test])
        hitung = rbf.fit(X_new[train], Y_new[train])
        predictrbf1 = rbf.predict(X_new[train])
        predictrbf = rbf.predict(X_new[test])
        fpr1, tpr1, _1 = metrics.roc_curve(Y_new[train], predictrbf1)
        fpr, tpr, _ = metrics.roc_curve(Y_new[test], predictrbf)
        auc_1 = metrics.auc(fpr1,tpr1)
        auc_ = metrics.auc(fpr,tpr)
        print("\nConfusiion Matrix Train: \n",
metrics.confusion_matrix(Y_new[train], predictrbf1))
        print("\nConfusiion Matrix Test: \n",
metrics.confusion_matrix(Y_new[test], predictrbf))
        akurasi.append(accuracy_score(Y_new[train], predictrbf1))
        akurasi.append(accuracy_score(Y_new[test], predictrbf))
        print("Akurasi Prediksi Train: ", accuracy_score(Y_new[train],
predictrbf1))
        print("Akurasi Prediksi Test: ", accuracy_score(Y_new[test], predictrbf))
        print()
        print(metrics.classification_report(Y_new[train], predictrbf1))
        print(metrics.classification_report(Y_new[test], predictrbf))
        print("AUC Train: ", auc_1)
        print("AUC Test: ", auc_)
```

Lampiran 7. Syntax SMOTE

```
data=read.csv("E:/KULIAH/TUGAS AKHIR LJ/FULL TWEET/DATA
UNTUK BRI TANPA REPLIES/datauntuksmoteBRI.csv", sep=",", dec=".")
X=subset(data,select=-c(y))
target=subset(data,select=c(y))
library(smotefamily)
K=3
dup_size=40
#-----------------------------------------------------------
ncD=ncol(X) #The number of attributes
n_target=table(target)
classP=names(which.min(n_target))
P_set=subset(X,target==names(which.min(n_target)))[sample(min(n_target)),]
#Extract a set of positive instances
N_set=subset(X,target!=names(which.min(n_target)))
P_class=rep(names(which.min(n_target)),nrow(P_set))
N_class=target[target!=names(which.min(n_target))]
sizeP=nrow(P_set) #The number of positive instances
sizeN=nrow(N_set) #The number of negative instances
knear=knearest(P_set,P_set,K)
sum_dup = n_dup_max(sizeP+sizeN,sizeP,sizeN,dup_size)
syn_dat=NULL
 for(i in 1:sizeP)
              {       if(is.matrix(knear))
              {    pair_idx = knear[i,ceiling(runif(sum_dup)*K)]
                                    }else
                                                {    pair_idx = rep(knear[i],sum_dup)
                                                }
              g = runif(sum_dup)
                                                P_i =
matrix(unlist(P_set[i,]),sum_dup,ncD,byrow=TRUE)
                                                Q_i = as.matrix(P_set[pair_idx,])
                                                syn_i = P_i + g*(Q_i - P_i)
                                                syn_dat = rbind(syn_dat,syn_i)
     }
P_set[,ncD+1] = P_class
colnames(P_set)=c(colnames(X),"class")
N_set[,ncD+1] = N_class
colnames(N_set)=c(colnames(X),"class")
rownames(syn_dat)= NULL
syn_dat = data.frame(syn_dat)
syn_dat[,ncD+1] = rep(names(which.min(n_target)),nrow(syn_dat))
colnames(syn_dat)=c(colnames(X),"class")
```

Lampiran 7. Syntax SMOTE (Lanjutan)

```
NewD=rbind(P_set,syn_dat,N_set)
newD = data.frame(NewD)
rownames(NewD) = NULL
D_result = list(data = NewD, syn_data = syn_dat, orig_N = N_set, orig_P =
P_set, K = K, K_all = NULL, dup_size = sum_dup, outcast = NULL, eps =
NULL, method="SMOTE")
class(D_result) = "gen_data"
return(D_result)

write.csv(syn_dat, "E:/KULIAH/TUGAS AKHIR LJ/FULL TWEET/DATA
UNTUK MANDIRI TANPA REPLIES/data sintetik smote1.csv")
write.csv(newD, "E:/KULIAH/TUGAS AKHIR LJ/FULL TWEET/DATA
UNTUK MANDIRI TANPA REPLIES/data smote1.csv")
```

Lampiran 8. Syntax *Wordcloud*

```
library(RColorBrewer)
library(wordcloud)
library(tm)

dataTFIDF = read.csv("E:/KULIAH/TUGAS AKHIR LJ/FULL
TWEET/DATA UNTUK BRI TANPA
REPLIES/tfidfbriunigram_posnothanks.csv",header=TRUE)
names(dataTFIDF)
head(dataTFIDF)
jumlahdata = rowSums(dataTFIDF)
gabTFIDF = data.frame(dataTFIDF, jumlahdata);gabTFIDF
m = ncol(gabTFIDF);m
TFIDFman = gabTFIDF[which(gabTFIDF$jumlah!=0),-348];TFIDFman

a = sort(colSums(TFIDFman), decreasing = TRUE);a
b = data.frame(word = names(a), freq=a);b
head(b)
win.graph()
wordcloud(words = b$word, freq=b$freq, scale = c(4,0,4), random_color =
FALSE,
        random.order = FALSE, rot.per = 0.25, min_freq = 3, max.words =
Inf,
        colors = brewer.pal(8, "Dark2"))
```

Lampiran 9. Syntax Estimasi Parameter untuk Regresi Logistik Biner

```
library(dplyr)
library(Hmisc)
library(pscl)
library(ResourceSelection)
library(caret)
library(polycor)
library(car)

data<-read.csv("E:/KULIAH/TUGAS AKHIR LJ/FULL TWEET/DATA
UNTUK MANDIRI TANPA REPLIES/data training lexicon
mandiri.csv",header=TRUE)
datakor<-read.csv("E:/KULIAH/TUGAS AKHIR LJ/FULL TWEET/DATA
UNTUK MANDIRI TANPA REPLIES/datakorelasi.csv",header=TRUE)
options(max.print = 999999999)
kor=data.frame(datakor)
View(data)

#Korelasi Polycor
hasil<-hetcor(kor, stde.err=TRUE, tests=TRUE)
write.csv(hasil[["correlations"]], "E:/KULIAH/TUGAS AKHIR LJ/FULL
TWEET/DATA UNTUK MANDIRI TANPA REPLIES/korelasiyx.csv")
write.csv(hasil[["tests"]], "E:/KULIAH/TUGAS AKHIR LJ/FULL
TWEET/DATA UNTUK MANDIRI TANPA REPLIES/pvalueyx.csv")
write.csv(hasil[["std.errors"]], "E:/KULIAH/TUGAS AKHIR LJ/FULL
TWEET/DATA UNTUK MANDIRI TANPA REPLIES/stderroryx.csv")
View(hasil)

#Korelasi Pearson
data.cor=rcorr(as.matrix(datakor))
write.csv(data.cor[["r"]], "E:/KULIAH/TUGAS AKHIR LJ/FULL
TWEET/DATA UNTUK MANDIRI TANPA REPLIES/korelasixx.csv")
write.csv(data.cor[["P"]], "E:/KULIAH/TUGAS AKHIR LJ/FULL
TWEET/DATA UNTUK MANDIRI TANPA REPLIES/pvaluexx.csv")

#Regresi Logistik Biner
model <- glm(Kode.Klasifikasi~., data=data, family=binomial)
summary(model)
```

Lampiran 10. Hasil Tuning Parameter

```
library(e1071)
data<-read.csv("E:/KULIAH/TUGAS AKHIR LJ/FULL TWEET/DATA UNTU
K MANDIRI TANPA REPLIES/FIX DATA MANDIRI/data smote.csv", sep=",
", dec=".", header=TRUE)
View(data)
ncol(data)
x=data[,-461]
y=data[,461]

para <- tune.svm(x,y,
                 cost =c(0.01,0.1,1,10,100),
                 #gamma = c(0.01,0.1,1,10,100),
                 kernel="linear")
para$best.mode

para1 <- tune.svm(x,y,
                 cost =c(0.01,0.1,1,10,100),
                 gamma = c(0.01,0.1,1,10,100),
                 kernel="radial")
para1$best.mode
-----------------------------------------------------
Call:
best.svm(x = x, y = y, cost = c(0.01, 0.1, 1, 10, 100), kernel = "linear")
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  linear
       cost:  0.01
      gamma:  0.002173913
    epsilon:  0.1
Number of Support Vectors:  2087
Call:
best.svm(x = x, y = y, gamma = c(0.01, 0.1, 1, 10, 100), cost = c(0.01, 0.1,
    1, 10, 100), kernel = "radial")
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  10
      gamma:  0.01
    epsilon:  0.1
Number of Support Vectors:  1955
```

Lampiran 10. Hasil Tuning Parameter (Lanjutan)

```
Bank BRI

Call:
best.svm(x = x1, y = y1, cost = c(0.01, 0.1, 1, 10, 100), kernel = "linear")
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  linear
       cost:  0.1
      gamma:  0.005376344
    epsilon:  0.1
Number of Support Vectors:  576

Call:
best.svm(x = x1, y = y1, gamma = c(0.01, 0.1, 1, 10, 100), cost = c(0.01, 0.1,
    1, 10, 100), kernel = "radial")
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  100
      gamma:  0.01
    epsilon:  0.1
Number of Support Vectors:  168
-----------------------------------------------------
```

Lampiran 11. Surat Pernyataan

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FMKSD ITS:

Nama : Erna Dwi Nurindah Sari

NRP : 06211745000034

menyatakan bahwa data yang digunakan dalam Tugas Akhir / Thesis ini merupakan data sekunder yang diambil dari penelitian / Buku / Tugas Akhir / Thesis / publikasi lainnya yaitu:

Sumber : Twitter API (*Application Program Interface*)

Keterangan : Data *tweet* dengan *keywords* "kontakBRI" dan "mandiricare"

Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui,
Pembimbing Tugas Akhir

(Irhamah, M.Si., Ph.D.)
NIP. 19780406 200112 2 002

Surabaya, 15 Juli 2019

(Erna Dwi Nurindah Sari)
NRP. 06211745000034

*(coret yang tidak perlu)

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Penulis bernama lengkap Erna Dwi Nurindah Sari lahir di Tulungagung, 27 September 1995 sebagai adik dari Farid Ashari, saudara kembar dari Erni Dwi Cahyati, dan kakak dari Lina Fachrun Nisa' dari pasangan Siswanto dan Sri Ismiati. Penulis bertempat tinggal di Ds. Sumberdadi, Kec. Sumbergempol, Kab. Tulungagung. Menempuh pendidikan formal di TK Perwanida Sumberdadi, SDN 1 Sumberdadi, SMPN 1 Sumbergempol, dan SMAN 1 Ngunut. Setelah lulus SMA penulis mengikuti seleksi penerimaan mahasiswa baru dan diterima di Jurusan Diploma III Statistika FMIPA ITS pada tahun 2014 dengan NRP yang dimiliki yakni 1314030083 dan melanjutkan S1 pada program Lintas Jalur di Statistika FMKSD ITS pada tahun 2017. Selama perkuliahan penulis aktif dalam organisasi kemahasiswaan, yakni sebagai staf Departemen Hubungan Luar di HIMADATA-ITS periode 2015/2016 dan sebagai kabinet Departemen Media Informasi di HIMADATA-ITS periode 2016/2017. Apabila pembaca memiliki kritik dan saran atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini, penulis dapat dihubungi melalui email erna.ernadwi@gmail.com.

Halaman ini sengaja dikosongkan