TUGAS AKHIR – KS184822

FEATURE SELECTION UNTUK PREDIKSI TELAT BAYAR PELANGGAN MENGGUNAKAN METODE *REGULARIZED* SVM DAN *REGULARIZED* REGRESI LOGISTIK

THALIA MARDA SANTIKA
NRP 062116 4000 0087

Dosen Pembimbing
Dr. rer. pol. Dedy Dwi Prastyo, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

(Halaman ini sengaja dikosongkan)

TUGAS AKHIR – KS184822

FEATURE SELECTION UNTUK PREDIKSI TELAT BAYAR PELANGGAN MENGGUNAKAN METODE *REGULARIZED SVM DAN REGULARIZED REGRESI LOGISTIK*

THALIA MARDA SANTIKA
NRP 062116 4000 0087

Dosen Pembimbing
Dr. rer. pol. Dedy Dwi Prastyo, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

(Halaman ini sengaja dikosongkan)

FINAL PROJECT – KS184822

FEATURE SELECTION FOR CUSTOMER'S PAYMENT LATE PREDICTION USING REGULARIZED SVM AND REGULARIZED LOGISTIC REGRESSION METHOD

THALIA MARDA SANTIKA
NRP 062116 4000 0087

Supervisor
Dr. rer. pol. Dedy Dwi Prastyo, S.Si., M.Si.

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF SCIENCE AND DATA ANALYTICS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

APLIKASI METODE BERBASIS QUANTILE DELTA MAPPING UNTUK KOREKSI BIAS PADA DATA EARTH SYSTEM MODEL PULAU JAWA

TUGAS AKHIR

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

Oleh :

Cahya Idznii Igawati
NRP. 062116 4000 0017

Disetujui oleh Pembimbing :

Dr.rer.pol. Heri Kuswanto, S.Si., M.Si.
NIP. 19820326 200312 1 004

M. Sjahid Akbar, S.Si., M.Si.
NIP. 19720705 199802 1 001

Mengetahui,
Kepala Departemen

Dr. Dra. Kartika Fithriasari M.Si
NIP. 19691212 199303 2 002

SURABAYA, JANUARI 2020

(Halaman ini sengaja dikosongkan)

FEATURE SELECTION UNTUK PREDIKSI TELAT BAYAR PELANGGAN MENGGUNAKAN METODE *REGULARIZED* SVM DAN *REGULARIZED* REGRESI LOGISTIK

Nama Mahasiswa : Thalia Marda Santika
NRP : 062116 4000 0087
Departemen : Statistika-FMKSD-ITS
Dosen Pembimbing : Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.

Abstrak

Telat bayar merupakan keadaan dimana sesorang yang tidak dapat memenuhi kewajibannya pada waktu yang telah ditentukan. Analisis telat bayar sangat penting dilakukan untuk mengukur tingkat risiko pelanggan. Penelitian ini bertujuan untuk memprediksi risiko telat bayar pada pelanggan dengan menggunakan metode klasifikasi. Support Vector Machine (SVM) dan regresi logistik adalah dua metode yang digunakan pada penelitian ini dimana dalam pengaplikasiannya juga menambahkan pendekatan metode regularisasi yaitu lasso dan elastic-net. Metode regularisasi digunakan untuk mengestimasi parameter pada model yang sekaligus dapat menyeleksi variabel sehingga didapatkan hasil seleksi variabel yang relevan. Adanya variabel input yang relevan diharapkan dapat menaikkan performansi model terutama ketika diterapkan pada data testing. Data yang digunakan pada penelitian ini adalah data sekunder yang merupakan data pembayaran pelanggan Perusahaan Telco selama enam bulan pada tahun 2018. Pelanggan akan diklasifikasikan menjadi dua kelas, yaitu pelanggan yang telat membayar dan tidak. Hasil penelitian menunjukkan penambahan regularisasi pada model dapat meningkatkan nilai akurasi. Lasso regresi logistik merupakan metode yang paling baik digunakan dengan nilai AUC sebesar 0,59640 dengan 33 variabel prediktor yang relevan. Hasil ini dipengaruhi oleh karakteristik data yang tidak dapat dengan baik membedakan klasifikasi antara pelanggan yang telat membayar maupun tidak.

Kata Kunci : Feature Selection, Telat Bayar Pelanggan, Klasifikasi, Regularized Regresi Logistik, Regularized SVM

(Halaman ini sengaja dikosongkan)

FEATURE SELECTION FOR CUSTOMER'S PAYMENT LATE PREDICTION USING REGULARIZED SVM AND REGULARIZED LOGISTIC REGRESSION METHOD

Name : **Thalia Marda Santika**
Student Number : **062116 4000 0087**
Department : **Statistics**
Supervisors : **Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.**

Abstract

Late payement is a condition where a person cannot fulfill its obligations in the agreed time. Analysis of late payment is important to do because it can measure the level of customer risk. This study aims to predict the risk of customer's late payment by using a classification method. Support Vector Machine (SVM) and logistic regression are two methods that used in this study where in the application also added the regularization method lasso and elastic-net. The regularization method is used to estimate parameters in the model that can simultaneously select variables so that the relevan variable selection results are obtained. The existence of relevant input variables is expected to improve the performance of the model, especially when applied to data testing. The data used in this study are secondary data which are data of Telco Company's customer payments for six months in 2018 by using variable payment behavior, total bills, and frequency of using services. Customers will be classified into two classes, customers who are late paying and never late to pay bills. The results showed the addition of regularization on the model can increase the value of accuracy. Logistic regression Lasso is the best method used with an AUC value of 0.59640 with 33 relevant predictor variables. This result is influenced by the characteristics of the data that cannot properly distinguish the classification between customers who are late paying or not.

Keywords: Feature selection, Payement Default, Klasifikasi, Regularized Regresi Logistik, Regularized SVM

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji Syukur penulis panjatkan atas berkat, rahmat, dan karunia yang telah diberikan Allah SWT sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini yang berjudul **"*Feature Selection* untuk Prediksi Telat Bayar Pelanggan Menggunakan Metode *Regularized* SVM dan *Regularized* Regresi Logistik "** dengan tepat waktu.

Penulis menyadari dalam penyusunan Tugas Akhir ini tidak akan selesai tanpa bantuan maupun dukungan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan terima kasih kepada:

1. Orang tua, adik dan keluarga penulis yang selalu memberikan doa dan dukungan selama penyusunan Tugas Akhir.
2. Dr. rer. pol. Dedy Dwi Prastyo, S.Si., M.Si. selaku dosen pembimbing yang telah memberikan bimbingan, saran, serta motivasi selama penyusunan Tugas Akhir berlangsung.
3. Santi Puteri Rahayu, S.Si., M.Si., Ph.D dan Dr. Dra. Kartika Fithriasari, M.Si selaku dosen penguji yang telah memberikan masukan dan bantuan dalam menyelesaikan Tugas Akhir.
4. Dr. Suhartono, S.Si., M.Sc selaku dosen wali yang telah banyak memberikan saran dan arahan dalam proses belajar selama ini di Departemen Statistika.
5. Dr. Dra. Kartika Fithriarsari M.Si selaku Kepala Departemen Statistika dan Vita Ratnasari, S.Si., M.Si dan Dr. Santi Wulan Purnami, S.Si., M.Si. selaku Sekretaris Departemen Statistika FSAD ITS.
6. Azaria Natasha, M.Si yang telah banyak membantu dalam pengerjaan Tugas Akhir dan mendapatkan data penelitian.
7. Sahabat penulis yaitu Laili, Cahya, Kinanthi, Riris, Fransiska, Inan, Erika dan Moniyca yang selalu memberikan support dan semangat dalam penyusunan Tugas Akhir.
8. Teman-teman seperjuangan TA khususnya Abid dan Ikayang menemani penulis dalam running data dan sebagai teman diskusi serta teman-teman TR16GER lainnya yang

selalu memberikan semangat kepada penulis dalam penyusunan Tugas Akhir.

9. Seluruh pihak yang turut membantu dalam penyelesaian laporan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Penulis menyadari masih banyak kekurangan dalam pembuatan laporan Tugas Akhir ini. Penulis berharap semoga laporan Tugas Akhir ini dapat bermanfaat dan menambah wawasan bagi pembaca. Kritik dan saran sangat diperlukan untuk perbaikan di masa yang akan datang.

Surabaya, Januari 2020

Penulis

DAFTAR ISI

DAFTAR GAMBAR

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pelanggan merupakan salah satu aset terpenting dalam perusahaan. Keberlangsungan sebuah perusahaan bergantung pada pelanggannya. Hal ini dikarenakan sumber pendapatan utama dari perusahaan diperoleh dari pembayaran pelanggan. Nantinya dari pendapatan tersebut, perusahaan dapat memperoleh laba yang digunakan untuk membiayai segala aktivitas operasional dan investasi perusahaan (Reimeinda, et al., 2016). Pentingnya peran dan posisi pelanggan membuat setiap perusahaan berusaha untuk mendapatkan pelanggan sebanyak-banyaknya. Namun semakin banyak pelanggan maka akan semakin banyak masalah yang dihadapi oleh perusahaan yang dapat menyebabkan tersendatnya pendapatan perusahaan. Ketika arus keuangan dalam bisnis bermasalah, *cash flow* perusahaan akan mejadi minus (Ye & Rahman, 2010). Salah satu permasalahan yang banyak ditemui pada pelanggan adalah terjadinya telat bayar.

Telat bayar merupakan salah satu jenis dari banyak macam *default* dalam konteks ekonomi. *Default* sendiri dapat diartikan sebagai ketidakmampuan seseorang atau perusahaan akibat tidak memiliki sumber daya keuangan untuk membayar kewajiban keuangan yang telah jatuh tempo pada waktunya (Laitinen, 2006). Biaya keuangan dan waktu pembayaran yang berkaitan dengan pembayaran yang lambat atau tidak pasti dapat mengikis profitabilitas penjualan (Wilson, 2008). Pentingnya kelancaran pembayaran pelanggan mendorong perusahaan untuk perlu mengetahui seberapa besar kemungkinan pelanggannya mengalami telat bayar dengan menggunakan prediksi telat bayar. Hal ini sangat penting digunakan sebagai bentuk dari manajemen risiko suatu perusahaan yang berguna untuk mengukur tingkat risiko pelanggan (Härdle, et al., 2014).

Penelitian awal mengenai *default analysis* dilakukan oleh Merwin (1942). Penelitian tersebut salah satunya membahas mengenai rasio pemberhentian dan keberlanjutan pada industri manufaktur. Kemudian munculah metode analisis diskriminan.

Beaver (1966) menggunakan analisis diskriminan univariat untuk menghitung rasio finansial pada perusahaan dan didapatkan bahwa arus kas terhadap total rasio hutang dapat memprediksi *default* pada perusahaan dengan akurasi paling tinggi. Sementara itu Altman (1968) melakukan pengembangan metode dengan menggunakan analisis diskriminan multivariat yang dapat menghasilkan nilai akurasi lebih tinggi daripada menggunakan metode analisis diskriminan univariat pada prediksi kebangkrutan perusahaan manufaktur. Penelitian ini menghasilkan akurasi sebesar 94% dengan menggunakan variabel modal, laba ditahan, penghasilan sebelum bunga dan pajak, penjualan yang keempatnya dihitung terhadap total aset dan nilai pasar dari ekuitas terhadap nilai buku dari total hutang. Pada tahun 1980 an analisis diskriminan mulai tergantikan dengan regresi logistik.

Regresi logistik adalah model terpenting untuk data respon kategorik dan banyak digunakan dalam berbagai aplikasi (Agresti, 2002). Regresi logistik juga banyak digunakan dalam ekonometrika, khususnya untuk analisis data finansial. Penelitian yang dilakukan oleh Ohlson (1980) bertujuan untuk memprediksi *default* perusahaan yang dibuktikan dengan terjadinya kebangkrutan metode menggunakan regresi logistik, dimana variabel yang digunakan adalah variabel akutansi seperti aset dan liabilitas perusahaan. Hasil penelitian ini membuktikan bahwa regresi logistik dapat menghasilkan prediksi *default* perusahaan yang lebih baik daripada menggunakan metode analisis diskriminan. Kelebihan dari model regresi logistik adalah tidak mengasumsikan normalitas multivariat dan matriks kovarians seperti yang pada analisis diskriminan. Model regresi logistik juga menggunakan fungsi distribusi kumulatif logistik dalam pemodelan probabilitas *default* (Chen, et al., 2006). Regresi logistik juga digunakan pada penelitian Laitinen (2006). Penelitian ini menggunakan analisis faktor dan analisis regresi PLS dalam data keuangan dari 3000 perusahaan untuk mengekstraksi faktor yang digunakan dalam analisis regresi logistik. Härdle & Prastyo (2014) melakukan prediksi *default* pada perusahaan di Asia Tenggara menggunakan model *regularized* regresi logistik. Penelitian ini menghasilkan akurasi yang sangat tinggi terutama

untuk industri Indonesia, Singapura dan Thailand dimana pemilihan variabel menggunakan metode *lasso* dan *elastic-net penalties*.

Selain regresi logistik, permasalahan mengenai *default analysis* banyak menggunakan metode analisis *Support Vector Machine* (SVM) dimana dapat menghasilkan akurasi yang lebih baik. SVM merupakan metode nonparametrik, multivariat yang dapat digunakan untuk klasifikasi linier maupun nonlinier dengan menggunakan pendekatan *supervised learning* (Natasha et al., 2019). Danenas, et al. (2011) melakukan penelitian yang membahas mengenai evaluasi risiko kredit menggunakan beberapa model *Support Vector Machines* (SVM) dengan analisis diskriminan dan *feature selection*. Hasil penelitian menunjukkan bahwa *liniear* SVM bersamaan dengan pengklasifikasian SVM berbasis gradien dan algoritma *Core Vector Machines*, dapat menjadi pilihan atau alternatif yang baik untuk implementasi model evaluasi risiko kredit berbasis SVM. Penelitian yang dilakukan oleh Härdle, et al. (2005) menyimpulkan bahwa SVM mampu mengekstraksi informasi dari data ekonomi pada kehidupan nyata dimana SVM mudah disesuaikan dengan hanya beberapa parameter. Hal ini membuat SVM sangat cocok digunakan untuk penilaian perusahaan dan metode penilaian risiko investasi yang diterapkan oleh lembaga keuangan. Prastyo (2015) melakukan penelitian mengenai prediksi *default* perusahaan dengan menggunakan *regularized* SVM dengan dua model yaitu *Least Absolute Shrinkage and Selection Operator* (Lasso) dan *Smoothly Clipped Absolutely Deviation* (SCAD). Penelitian ini menunjukkan kedua metode tersebut menghasilkan nilai akurasi yang sama, namun prediktor yang terpilih dengan menggunakan metode SVM-Lasso dapat memberikan interpretasi yang masuk akal.

Natasha (2019) menggunakan data Perusahaan Telco untuk menganalisis telat bayar pada pelanggan dengan membandingkan beberapa metode yaitu analisis diskriminan, regresi logistik biner, *Artificial Neural Network* (ANN), dan SVM, *Deep Neural Network* (DNN) dan *Deep Support Vector Learning* (DVSL). Hasil penelitian ini menunjukkan bahwa metode terbaik yang dapat

dapat digunakan untuk memprediksi telat bayar pelanggan adalah *Deep Neural Network* dengan nilai akurasi sebesar 0,73. Namun pada metode SVM, hasil prediksi pada data *training* menghasilkan nilai AUC sebesar 1. Namun prediksi telat bayar pada data *testing* menghasilkan nilai AUC sebesar 0,5. Model SVM yang didapat pada metode ini menunjukkan adanya *overfitting*. Hal ini diduga karena parameter yang didapatkan belum memberikan hasil yang optimum sehingga diperlukan pemilihan prediktor yang relevan untuk meningkatkan akurasi hasil prediksi.

Pemilihan prediktor atau lebih dikenal dengan istilah *feature selection*. *Feature selection* adalah metode yang digunakan untuk mengurangi dimensi dengan memilih subset dari variabel input asli. *Feature selection* banyak digunakan pada data yang memiliki puluhan bahkan ratusan variabel. Manfaat dari *feature selection* diantaranya adalah mempermudah visualisasi data dan pemahaman data, mengurangi persyaratan pengukuran dan penyimpanan, mengurangi waktu dalam pemodelan data, dan meningkatkan kinerja prediksi (Guyon & Elisseeff, 2003). Masalah yang paling sering dihadapi pada pemodelan data dengan dimensi tinggi adalah pemilihan model optimal. Pemodelan ini dapat mengurangi pilihan subset yang harus dimasukkan ke dalam model. Salah satu metode yang dapat digunakan pada kasus ini adalah *regularization* dimana metode ini melakukan pemilihan variabel dengan meminimalkan fungsi tujuan (Su & Zhang, 2014).

Berdasarkan latar belakang yang telah disebutkan, maka akan dilakukan analisis dengan memprediksi risiko telat bayar pada pelanggan Perusahaan Telco menggunakan metode *regularized* SVM dan *regularized* regresi logistik dimana dengan pendekatan metode regularisasi, Metode ini telah mencakup estimasi parameter dan *feature selection* yang dilakukan dalam satu tahapan sekaligus. Penelitian ini menggunakan data pembayaran pelanggan ke layanan telekomunikasi yang diberikan dengan analisis yang didasarkan pada perilaku pembayarannya, jumlah tagihan, dan frekuensi menggunakan layanan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas permasalahan yang muncul pada penelitian ini adalah bagaimana

memprediksi risiko telat bayar pada pelanggan Perusahaan Telco dengan mengunakan metode *regularized* SVM dan *regularized* regresi logistik. Penggunaan metode SVM linier dan regresi logisti juga digunakan sebagai pembanding antara metode dengan *feature selection* dan tanpa *feature selection*. Eksplorasi data dilakukan sebelum analisis untuk mengetahui karakteristik data. Analisis prediksi telat bayar pada pelanggan dilakukan untuk mengetahui klasifikasi pembayaran yang dilakukan dimana pelanggan akan masuk ke dalam kategori telat dan tidak pernah telat membayar. Sebagian besar metode klasifikasi biasanya menggunakan banyak prediktor, namun beberapa prediktor di dalamnya sebenernya tidak berpengaruh signifikan ataupun ada beberapa prediktor yang berkorelasi tinggi dengan varibel lain. Pada penelitian ini juga dilakukan *feature selection* untuk memilih variabel yang relevan dalam memprediksi kasus telat bayar pada pelanggan Perusahaan Telco. Hasil prediksi dari dua metode tersebut nantinya akan dibandingkan untuk mengetahui metode mana yang paling baik

1.3 Tujuan

Berdasarkan rumusan masalah, adapun tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut.

1. Mendeskripsikan karakteristik data.
2. Memprediksi risiko telat bayar pada pelanggan Perusahaan Telco dengan menggunakan SVM linier dan *regularized* SVM.
3. Memprediksi risiko telat bayar pada pelanggan Perusahaan Telco dengan menggunakan regresi logistik dan *regularized* regresi logistik.
4. Mendapatkan prediktor yang relevan dalam memprediksi risiko telat bayar berdasarkan metode terbaik.

1.4 Manfaat Penelitian

Manfaat yang diharapkan pada penelitian ini adalah sebagai berikut :

1. Bagi Keilmuan Statistika
   Dapat menjadi referensi untuk penelitian selanjutnya mengenai analisis telat bayar, khususnya pada kasus telat bayar pelanggan dengan pendekatan metode statistik. Selain

itu penggunaan metode *regularized* SVM dan *regularized* regresi logistik dapat dijadikan wawasan dan pengetahuan mengenai pengembangan metode SVM dan Regresi Logistik.

2. Bagi Perusahaan
   Dapat memberikan informasi mengenai seberapa besar probabilatas telat bayar pada pelanggannya. Penelitian ini juga dapat memberikan saran serta rekomendasi kepada perusahaan mengenai faktor-faktor yang signifikan dalam mempengaruhi telat bayar pada pelanggan. Model yang dihasilkan dapat dijadikan sebagai acuan dalam memprediksi telat bayar pelanggan pada waktu yang akan datang, sehingga perusahaan dapat menerapkan kebijakan yang dapat mengurangi telat bayar pada pelanggannya.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah metode *feature selection* yang digunakan adalah *Least Absolute Shrinkage and Selection Operator* (Lasso) dan *Elastic-Net*. Pengaruh variabel lain seperti faktor pribadi pelanggan diabaikan. Hasil analisis berfokus untuk mendapatkan variabel yang relevan didasarkan pada metode *regularized*. Variabel kategorik ditreatment menjadi variabel kontinyu pada metode *regularized* regresi logistik karena keterbatasan metode.

BAB II
TINJAUAN PUSTAKA

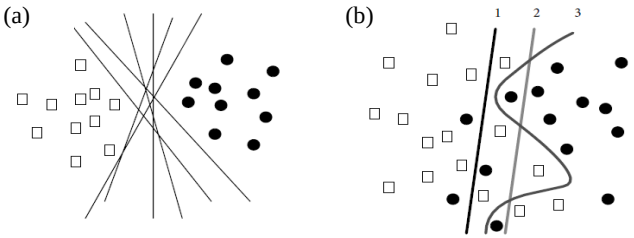
Tinjauan pustaka berisi landasan teori yang dipakai pada penelitian ini.Teori yang digunakan pada penelitian ini berasal dari buku, jurnal ilmiah, dan beberapa penelitian sebelumnya.

2.1 Klasifikasi

Klasifikasi digunakan untuk memprediksi masalah dengan menghasilkan pengklasifikasi individu terhadap kelas tertentu (Breiman, et al., 1984). Kriteria penting dalam prosedur klasifikasi yang baik adalah dapat menghasilkan pengklasifikasi yang akurat (dalam batas data) dan juga dapat memberikan wawasan dan pemahaman ke dalam struktur prediksi data. Masalah klasifikasi termasuk dalam kategori *supervised learning*.

2.2 Support Vector Machines (SVM)

Support Vector Machines (SVM) pertama kali dikenalkan oleh Vapnik pada tahun 1992. SVM dapat diterapkan secara luas baik untuk masalah klasifikasi, peramalan, dan estimasi (Wu & Wang, 2013). Cara kerja dari metode SVM adalah dengan mencari *optimum hyperplane* yang dapat memaksimalkan marjin atau jarak antara kelas data. SVM dibagi menjadi dua yaitu SVM linier dan SVM nonlinier. *Linear classifier* merupakan klasifikasi dengan batas keputusan linier sedangkan *nonlinear classifier* merupakan klasifikasi dengan batas keputusan bergantung pada data secara nonlinier. Klasifikasi linier SVM dibagi menjadi dua yaitu *linearly separable* dan *linearly nonseparable*.



Gambar 2.1 Klasifikasi Linier SVM (a*) linearly separable* dan (b) *linearly nonseparable* (Sumber: Härdle, et al., 2014)

2.2.1 SVM *in Linearly Separable*

Setiap observasi terdiri dari pasangan p prediktor $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip}) \in \mathbb{R}^p$ dengan $i = 1, 2, \dots, n$ dimana n adalah banyaknya observasi. Sementara $y_i \in \boldsymbol{y} = \{-1, 1\}$ adalah label atau kelas dari data tersebut. Apabila $\boldsymbol{x}_i$ merupakan anggota dari kelas 1, maka $\boldsymbol{x}_i$ mempunyai label $\boldsymbol{y}_i$= 1, begitu pula sebaliknya. Data yang diberikan merupakan himpunan data *training* dari kedua kelas berupa data pasangan yang akan diklasifikasi. Himpunan tersebut dapat ditulis daam persamaan berikut (Härdle, et al., 2014).

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\} \in \mathbb{R}^p \text{ x } \{-1, 1\}. \tag{2.1}$$

D_n berisi informasi untuk memprediksi y pada observasi baru. Label atau kelas y_i pada data *training* disebut *trainor* atau supervisor. Konsep utama yang digunakan untuk menetapkan pemisah linier adalah *dot product*. Keluarga F dari fungsi klasifikasi yang terdapat pada ruang data adalah sebagai berikut.

$$F = \{\mathbf{x}_i^T \mathbf{w} + b, \mathbf{w} \in \mathbb{R}\}, \tag{2.2}$$

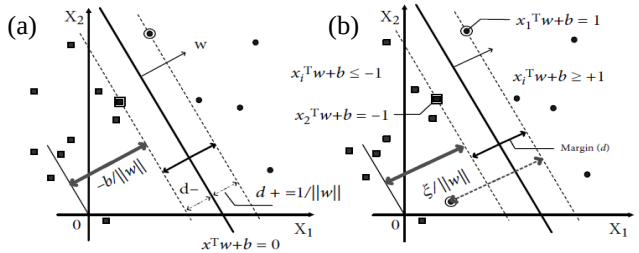
dimana $\boldsymbol{w}$ adalah vektor bobot yang tegak lurus terhadap *hyperplane* dan b merupakan posisi posisi bidang relatif terhadap pusat koordinat yang dapat ditunjukkan pada Gambar 2.2. Persamaan *hyperplane* pemisah dapat dituliskan sebagai berikut.

$$f(x_i) = \mathbf{x}_i^T \mathbf{w} + b = 0, \tag{2.3}$$

dimana

$\mathbf{x}_i^T \mathbf{w} + b \geq 1$ untuk $y_i = 1$ adalah bidang pembatas kelas 1. (2.4)

$\mathbf{x}_i^T \mathbf{w} + b \leq -1$ untuk $y_i = -1$ adalah bidang pembatas kelas 2. (2.5)



Gambar 2.2 *Hyperplane* Klasifikasi Linier SVM (a) *linearly separable*

dan (b) *linearly nonseparable* (Sumber: Härdle, et al., 2014)

Gambar 2.2 dapat ditunjukkan *hyperplane* memisahkan ruang menjadi dua bagian. Panjang vektor ***w*** adalah $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$. Bidang pembatas pertama dan kedua mempunyai bobot ***w*** dimana jarak tegak lurus dari titik asal adalah sebesar $\frac{|1-b|}{\|\mathbf{w}\|}$ dan $\frac{|-1-b|}{\|\mathbf{w}\|}$. Sementara itu jarak antara *margin* dengan *hyperplane* (bidang pemisah) adalah $d_+ = d_- = \frac{1}{\|\mathbf{w}\|}$. Sehingga didapatkan nilai maksimum margin (berdasarkan rumus jarak garis ke titik pusat) adalah $\frac{1-b-(-1-b)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$ atau ekuivalen dengan $\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|$ (Härdle, et al., 2014).

Menentukan *hyperplane* terbaik dapat menggunakan *Quadratic Programming* (QP) *problem* yaitu mencari titik minimal persaman (2.6) dengan memperhatikan *constraint* persamaan (2.7).

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|, \tag{2.6}$$

$$y_i(\mathbf{x}_i^T\mathbf{w} + b) - 1 \geq 0. \tag{2.7}$$

Lagrangian untuk masalah mendasar pada kasus ini adalah sebagai berikut.

$$\min_{\mathbf{w},b} L_p(\mathbf{w},b) = \min_{\mathbf{w},b}\left[\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\alpha_i(y_i(\mathbf{x}_i^T\mathbf{w}+b)-1)\right]. \tag{2.8}$$

Meminimalkan L terhadap ***w*** dan *b* dapat diberikan sebagai berikut.

$$\begin{aligned} \frac{\partial L_p(\mathbf{w},b)}{\partial \mathbf{w}} &= 0, \\ \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i x_i &= 0, \\ \mathbf{w} &= \sum_{i=1}^{n}\alpha_i y_i x_i, \end{aligned} \tag{2.9}$$

$$\frac{\partial L_p(\mathbf{w},b)}{\partial b} = 0,$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{2.10}$$

Lagrangian untuk permasalahan ganda dengan mensubtitusi persamaan (2.9) dan (2.10) adalah sebagai berikut.

$$\max_{\alpha} L_D(\alpha) = \max_{\alpha} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right]. \tag{2.11}$$

Setelah menyelesaikan permasalahan berganda, seseorang dapat mengklasifikasikan suatu objek dengan aturan klasifikasi sebagai berikut.

$$\hat{f}(x_i) = sign(\mathbf{x}_i^T \hat{\mathbf{w}} + b), \tag{2.12}$$

dimana $\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i$ dengan $\mathbf{x}_i$ adalah *support vector*. Nilai $b = \frac{1}{n_{SV}} \left(\sum_{i=1}^{n_{sv}} \frac{1}{y_i} - \left(\mathbf{x}_i^T \mathbf{w} \right) \right)$ dengan $\mathbf{x}_i^T$ merupakan data yang diklasifikasikan dan n_{sv} adalah jumlah *support vector*.

2.2.2 SVM *in Linearly Nonseparable*

Data pada kasus *linearly nonseparable* biasanya tidak dapat terpisah secara sempurna sehingga perlu diubah menjadi linier dengan menambahkan variabel *slack* ξ. Penambahan variabel ini menunjukkan pelanggaran terhadap ketelitian pemisah yang memungkinkan suatu titik berada di dalam *error margin* $0 \le \xi_i \le 1$ atau disebut misklasifikasi, $\xi > 1$ sehingga klasifikasi $\boldsymbol{x}_i$ adalah sebagai berikut (Härdle, et al., 2014).

$$\begin{aligned} &\mathbf{x}_i^T \mathbf{w} + b \ge 1 - \xi_i, \quad \text{untuk } y_i = 1, \\ &\mathbf{x}_i^T \mathbf{w} + b \ge -(1 - \xi_i), \text{ untuk } y_i = -1, \\ &\xi_i \ge 0, \end{aligned} \tag{2.13a}$$

dimana persamaan di atas dapat digabung menjadi,

$$y_i = (\mathbf{x}_i^T \mathbf{w} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \tag{2.13b}$$

Penalti untuk kesalahan klasifikasi terkait dengan jarak titik kesalahan klasifikasi x_i dari *hyperplane* yang membatasi kelasnya. Jika $\xi_i > 0$, maka kesalahan dalam memisahkan dua set terjadi. Fungsi obyektif sesuai dengan maksimalkan *penalized margin* kemudian diformulasikan sebagai berikut.

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i, \tag{2.14}$$

dengan nilai $\xi_i \geq 0$ dan parameter pinalti $C > 0$ dimana C adalah parameter yang menentukan besar kecilnya bobot akibat *misclassification* yang nilainya ditentukan.

Fungsi *Lagrange* dalam kasus ini adalah sebagai berikut.

$$L_p(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{x}_i^T \mathbf{w} + b) - 1 + \xi_i) - \sum_{i=1}^{n} \mu_i \xi_i, \tag{2.15}$$

dimana $\alpha_i \geq 0$ dan $\mu_i \geq 0$ merupakan *Lagrange Multiplier*. Nilai optimal dari persamaan (2.15) dapat dihitung dengan meminimalkan L terhadap $\mathbf{w}$, b dan ξ serta memaksimalkan L terhadap α sehingga diperoleh persamaan sebagai berikut.

$$\max_{\alpha} L_p(\alpha) = \max_{\alpha}(\min_{\mathbf{w},b,\xi} L(\mathbf{w}, b, \boldsymbol{\xi})). \tag{2.16}$$

Meminimalkan L terhadap $\mathbf{w}, b, \text{dan } \boldsymbol{\xi}$ dapat dilihat pada persamaan berikut.

$$\frac{\partial L_p(\mathbf{w}, b, \boldsymbol{\xi})}{\partial \mathbf{w}} = 0,$$

$$\mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i x_i = 0, \tag{2.17}$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i x_i,$$

$$\frac{\partial L_p(\mathbf{w}, b, \boldsymbol{\xi})}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0, \tag{2.18}$$

$$\frac{\partial L_p(\mathbf{w},b,\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = 0,$$
$$\frac{\partial L_p(\mathbf{w},b,\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = C - \alpha_i - \mu_i = 0, \tag{2.20}$$
$$\alpha_i = C - \mu_i.$$

Setelah itu dapat dilakukan subtitusi persamaan yang telah diturunkan ke dalam persamaan (2.15)

$$\max_{\alpha} L_D(\alpha) = \max_{\alpha}\left(\sum_{i-0}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right). \tag{2.21}$$

Constraint yang digunakan untuk memaksimalkan α_i pada persamaan (2.21) adalah sebagai berikut.

$$\sum_{i=1}^{n} \alpha_i y_i = 0,\ 0 \le \alpha_i \le C. \tag{2.22}$$

Sampel $\mathbf{x}_i$ untuk $\alpha_i > 0$ (*support vector*) merupakan titik yang berada di atas marjin atau di dalam marjin ketika *soft margin* mungkin digunakan. *Support vector* sering menyebar dan level penyebarannya berada pada batas atas untuk *misclassification rate* (Schölkopf & Smola, 2002).

2.3 *Regularized* SVM

Bentuk *regularized* merupakan bentuk pemisah secara linier. Persamaan *regularized* adalah penambahan *penalty function* pada *loss function* SVM. Standard L_2*-norm* SVM dapat ditulis dalam bentuk regularisasi seperti berikut (Prastyo, 2015).

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \{1 - y_i f(x_i)\}_+ + \lambda \|\mathbf{w}\|_2^2, \tag{2.23}$$

dimana $\lambda > 0$ adalah *tuning* parameter. *Regularized parameter* yaitu λ mempunyai nilai yang berbanding terbalik dengan C pada persamaan (2.14), artinya nilai λ sama dengan nilai $1/C$ (Hastie, et al., 2004). *Hinge-loss function* adalah sebagai berikut.

$$L\{y_i, f(x_i)\} = \{1 - y_i f(x_i)\}_+ \\ = \max\{0, 1 - y_i f(x_i)\}. \quad (2.24)$$

Penalty function pada persamaan (2.23) adalah $R(\mathbf{w}) = \|\mathbf{w}\|_2^2$ menggunakan semua input untuk membangun pemisah sehingga tidak dapat memilih prediktor yang relevan. *Regularization* SVM menggunakan istilah penalti lain yang dapat memilih prediktor yang relevan bersama dengan langkah-langkah estimasi parameter.

2.3.1 *Lasso* SVM

Salah satu teknik yang digunakan metode *regularized* adalah *Lasso*. Teknik Lasso diperkenalkan oleh Tibshirani (1996). *Lasso* adalah teknik membangun model populer yang secara bersamaan dapat menghasilkan model yang akurat dan bersifat *parsiminous*. Bradley & Mangasarian (1998) dan Zhu, et al. (2004) mengaplikasikan *Lasso* pada SVM dengan menggunakan istilah L_1*-norm* sebagai penalti sebagai ganti dari istilah awal yaitu L_2*-norm*.

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \{1 - y_i f(x_i)\}_+ + \lambda \|\mathbf{w}\|_1, \quad (2.25)$$

dimana besarnya $\lambda > 0$ merupakan *tuning parameter* (Becker, et al., 2011).

2.3.2 *Elastic-Net* SVM

Elastic-Net diperkenalkan oleh Zou & Hastie (2005) yang digunakan untuk mengatasi kelemahan *Lasso*. *Elastic-Net* adalah campuran dari penalti L_1 *norm* dan penalti L_2 *norm*. Kelebihan yang dimiliki *Elastic-Net* adalah jumlah prediktor yang dipilih tidak dibatasi oleh ukuran sampel dan kelompok prediktor yang berkorelasi dapat dipilih bersama-sama (pemilihan kelompok). Penalti L_1 berperan dalam pemilihan prediktor, sedangkan penalti L_2 berperan dalam pemilihan grup. Fungsi *Elastic-Net* SVM adalah sebagai berikut.

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \{1 - y_i f(x_i)\}_+ + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2, \quad (2.26)$$

dengan $\lambda_1, \lambda_2 \geq 0$ adalah *tuning parameter* (Becker, et al., 2011).

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* atau dikotomus dengan variabel prediktor (*x*) yang bersifat polikotomus (Hosmer dan Lemeshow, 1989). Regresi logistik biner merupakan regresi dengan variabel respon yang mempunyai dua kategori atau dua kejadian, yakni sukses (y = 1) atau gagal (*y* = 0). Fungsi kepadatan probabilitas untuk setiap pasangan adalah sebagai berikut:

$$f(y_i) = \left[\pi(\mathbf{x}_i)^{y_i}\right]\left[1-\pi(\mathbf{x}_i)\right]^{1-y_i}; y = 0,1, \tag{2.27}$$

dimana jika *y* = 0 maka $f(y) = (1-\pi)$ dan jika *y* = 1 maka $P(y=1) = \pi$. Model regresi logistik dengan prediktor $x_1, x_2, ..., x_p$ dapat dituliskan sebagai berikut:

$$\pi(\mathbf{x}_i) = \frac{e^{(\beta_0+\beta_1 x_1+...+\beta_p x_p)}}{1+e^{(\beta_0+\beta_1 x_1+...+\beta_p x_p)}}, \tag{2.28}$$

Estimasi parameter yang digunakan untuk menaksir parameter **β** pada model regresi logistik biner adalah metode *Maximum Likelihood Estimation* (MLE). Fungsi *liklihood* dapat diperoleh sebagai berikut:

$$l(\boldsymbol{\beta}) = \prod_{i=1}^{n} \pi(\mathbf{x}_i)^{y_i}\left[1-\pi(\mathbf{x}_i)\right]^{1-y_i}. \tag{2.29}$$

Fungsi likelihood tersebut lebih mudah dimaksimumkan dalam bentuk log dimana,

$$\begin{aligned} L(\boldsymbol{\beta}) &= \ln l(\boldsymbol{\beta}) \\ &= \sum_{i=1}^{n}\left\{y_i \ln\left[\pi(\mathbf{x}_i)\right] + (1-y_i)\ln\left[1-\pi(\mathbf{x}_i)\right]\right\}. \end{aligned} \tag{2.30}$$

Nilai **β** didapatkan dari hasil diferensial $L(\boldsymbol{\beta})$ terhadap β_j dan hasilnya adalah sama dengan nol.

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n}\sum_{j=0}^{p} y_i x_{ij} - \sum_{i=1}^{n} \frac{e^{\sum_{j=0}^{p}\beta_j x_{ij}}}{1+e^{\sum_{j=0}^{p}\beta_j x_{ij}}}\left(\sum_{j=0}^{p} x_{ij}\right) \tag{2.31}$$

sehingga,

$$\sum_{i=1}^{n}\sum_{j=0}^{p} y_i x_{ij} - \sum_{i=1}^{n} \pi(\mathbf{x}_i)\left(\sum_{j=0}^{p} x_{ij}\right) = 0. \qquad (2.32)$$

Estimasi varians kovarians dikembangkan melalui teori MLE dari koefisien parameternya yang didapatkan dari turunan kedua $l(\boldsymbol{\beta})$.

$$\frac{\partial^2 L(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_j'} = \sum_{i=1}^{n}\sum_{j=0}^{p} x_{ij} x_{ij}' \pi(\mathbf{x}_i)\left(1-\pi(\mathbf{x}_i)\right) \qquad (2.33)$$

Nilai taksiran $\boldsymbol{\beta}$ dari turunan pertama fungsi $L(\boldsymbol{\beta})$ yang nonlinier didapatkan dengan menggunakan metode iterasi Newton Raphson dengan rumus sebagai berikut:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left(\mathbf{H}\left(\boldsymbol{\beta}^{(t)}\right)\right)^{-1} \mathbf{g}(\boldsymbol{\beta})^{(t)}, t = 1, 2, \ldots, n. \qquad (2.34)$$

Iterasi dilakukan sampai konvergen ke-n. $\mathbf{H}\left(\boldsymbol{\beta}^{(t)}\right)$ merupakan matriks Hessian dengan

$$\mathbf{H}\left(\boldsymbol{\beta}^{(t)}\right) = \begin{bmatrix} h_{00} & h_{01} & \cdots & h_{0p} \\ h_{10} & h_{11} & \cdots & h_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ h_{p0} & h_{p1} & \cdots & h_{pp} \end{bmatrix}, h_{ij} = \frac{\partial^2 L(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_j'} \qquad (2.35)$$

dan $\mathbf{g}(\boldsymbol{\beta})^{(t)}$ merupakan vektor gradient dimana

$$\mathbf{g}(\boldsymbol{\beta})^{(t)} = \left[\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_0}, \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_1}, \ldots, \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_p}\right]. \qquad (2,36)$$

Matrix Hessian dengan iterasi sebanyak n dapat ditulis menjadi

$$\mathrm{H}\left(\beta^{(t)}\right) = -\left\{\mathbf{x}^T diag\left[\pi(\mathbf{x}_1)^{(t)}\left(1-(\mathbf{x}_1)^{(t)}\right), \ldots, \pi(\mathbf{x}_n)^{(t)}\left(1-(\mathbf{x}_n)^{(t)}\right)\right]\mathbf{x}\right\}^{-1} \qquad (2.37)$$

Sehingga didapatkan estimasi sebagai berikut:

$$\hat{\boldsymbol{\beta}}^{(t+1)} = \hat{\boldsymbol{\beta}}^{(t)} + \left\{ \mathbf{x}^T diag\left[\pi\left(\mathbf{x}_1\right)^{(t)} \left(1-\left(\mathbf{x}_1\right)^{(t)}\right),...,\pi\left(\mathbf{x}_n\right)^{(t)} \left(1-\left(\mathbf{x}_n\right)^{(t)}\right)\right]\mathbf{x}\right\}^{-1} \mathbf{x}^T\left(y-\pi\left(\mathbf{x}_n\right)^{(t)}\right) \tag{2.38}$$

Langkah-langkah iterasi Newton Raphson adalah sebagai berikut:

1. Menentukan nilai awal dari $\hat{\boldsymbol{\beta}}$ pada saat iterasi pertama yaitu $\hat{\boldsymbol{\beta}} = 0.$
2. Mulai dari iterasi pertama atau *t*=0 dilakukan iterasi dengan menghitung persamaan (2.44).
3. Iterasi dilakukan sampai konvergen dengan pegecekan dimana $\left\|\hat{\boldsymbol{\beta}}^{(t+1)} - \hat{\boldsymbol{\beta}}^{(t)}\right\| \leq \Theta$, dimana nilai Θ sangat kecil.

2.4.1 Uji Serentak

Pengujian parameter secara serentak dilakukan dengan menggunakan uji *likelihood* atau sering disebut dengan *Likelihood Ratio Test*. Uji ini merupakan uji *Chi-Squared* yang menggunakan nilai *maximum likelihood*. Uji ini bertujuan untuk memeriksa apakah variabel independen berpengaruh secara signifikan terhadap variabel depeden. Hipotesis dari pengujian serentak ini adalah sebagai berikut:

H_0: $\beta_1 = \beta_2 = ... = \beta_p = 0$

H_1: minimal ada satu $\beta_j \neq 0,$ dengan $j = 1,2,...,p.$

Statistik uji yang digunakan adalah sebagai berikut:

$$G = -2\ln\left[\frac{\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n_0}{n}\right)^{n_0}}{\prod_{i=1}^{n}\hat{\pi}_i^{y_i}\left(1-\hat{\pi}_i\right)^{1-y_i}}\right], \tag{2.39}$$

dimana $n_1 = \sum y_i$ dan $n_0 = \sum (1 - y_i)$. Statistik uji *G* merupakan *Likelihood Ratio Test* dimana nilai *G* mengikuti distribusi *Chi-Squared* dengan derajat bebas *p*, sehingga H_0 ditolak apabila $G > \chi^2_{(p,\alpha)}$ atau *p-value* < α (Hosmer dan Lemeshow, 2000).

2.4.2 Uji Parsial

Pengujian secara parsial dilakukan untuk mengetahui signifikansi masing-masing parameter terhadap variabel dependen. Pengujian parameter secara parsial menggunakan uji *Wald* dengan hipotesis sebagai berikut:

H_0: $\beta_j = 0$

H_1: $\beta_j \neq 0$, dengan $j = 1, 2, ..., p$.

Statistik uji untuk uji parsial adalah sebagai berikut:

$$W = \frac{\hat{\beta}_j}{\widehat{SE}(\hat{\beta}_j)} \tag{2.40}$$

Statistik uji *W* disebut juga statistik uji *Wald* dengan $\widehat{SE}(\hat{\beta}_j)$ adalah taksiran standar *error* parameter. H_0 ditolak jika $|W| > Z_{\alpha/2}$ atau *p-value* < α (Hosmer dan Lemeshow, 2000).

2.4.3 *Odd Ratio*

Menurut Hosmer dan Lemeshow (2000), estimasi koefisien dari variabel prediktor menyatakan slope atau nilai perubahan variabel dependen untuk setiap perubahan satu unit variabel independen. Intepretasi meliputi penentuan hubungan fungsional antara variabel dependen dan variabel independen serta mendefinisikan unit perubahan variabel dependen yang disebabkan oleh variabel independen. Koefisien parameter diinterpretasi dengan menggunakan nilai *odds ratio* (ψ). *Odd ratio* yang bernilai e^{β} untuk variabel kategorik dan $e^{c\beta}$ untuk variabel kontinyu dimana *c* merupakan unit perubahan. *Odd ratio* diartikan sebagai kecenderungan variabel respon memiliki suatu nilai tertentu jika diberikan *x*=1 dan dibandingkan pada *x*=0. Keputusan

tidak terdapat hubunga antara variabel prediktor dengan variabel respon diambil jika nilai *odds ratio* $(\psi) = 1$.

2.5 Regularized Regresi Logistik

Probabilitas telat bayar untuk pelanggan ke-*i* yang diberikan oleh prediktor telat bayar x_i dapat dirumuskan sebagai berikut (Härdle & Prastyo, 2014).

$$\begin{aligned} P\left(y_i = 1 \middle| x_i\right) &= \frac{e^{\beta_0 + x_i^T \boldsymbol{\beta}}}{1 + e^{\beta_0 + x_i^T \boldsymbol{\beta}}} \\ &= \frac{1}{1 + e^{-(\beta_0 + x_i^T \boldsymbol{\beta})}}, \end{aligned} \tag{2.41}$$

dan

$$\begin{aligned} P\left(y_i = 0 \middle| x_i\right) &= 1 - P\left(y_i = 1 \middle| x_i\right) \\ &= \frac{1}{1 + e^{(\beta_0 + x_i^T \boldsymbol{\beta})}}. \end{aligned} \tag{2.42}$$

Log *odds ratio* adalah model regresi linier dimana,

$$\log\left\{\frac{P\left(\gamma_1 = 1 \middle| x_i\right)}{P\left(\gamma_1 = 0 \middle| x_i\right)}\right\} = \beta_0 + x_i^T \boldsymbol{\beta}. \tag{2.43}$$

Kemudian dilanjutkan dengan memaksimalkan fungsi *regularized log-likehood*.

$$\max_{\beta, \beta_0} \left\{\ell\left(\beta_0, \boldsymbol{\beta}\right) - \lambda R_\gamma\left(\boldsymbol{\beta}\right)\right\}, \tag{2.44}$$

dengan $R_\gamma\left(\boldsymbol{\beta}\right)$ adalah fungsi regularisasi dan likelihoodnya adalah sebagai berikut.

$$\begin{aligned} \ell\left(\beta_0, \boldsymbol{\beta}\right) &= n^{-1} \sum_{i=0}^{n} \left[y_i \log P\left(y_i = 1 \middle| x_i\right) + \left(1 - y_i\right) \log\left\{1 - P\left(y_i = 1 \middle| x_i\right)\right\} \right] \\ &= n^{-1} \sum_{i=0}^{n} \left\{ I\left(y_i = 1\right) \log P\left(y_i = 1 \middle| x_i\right) \right\} + \left\{ I\left(y_i = 0\right) \log P\left(y_i = 0 \middle| x_i\right) \right\} \\ &= n^{-1} \sum_{i=0}^{n} I\left(y_i = 1\right) \left[\left(\beta_0 + x_i^T \boldsymbol{\beta}\right) - \log\left\{1 + e^{\beta_0 + x_i^T \boldsymbol{\beta}}\right\} \right]. \end{aligned} \tag{2.45}$$

Persamaan. (2.43) dimaksimalkan menggunakan algoritma penurunan koordinat titik. Tiga langkah dalam algoritma penurunan koordinat siklik yaitu lingkaran luar, tengah, dan dalam. Nilai λ diatur dan menghasilkan $\left(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}\right)$ pada loop luar. Perkiraan kuadratik (ekspansi Taylor) dari fungsi log-likelihood diperbarui pada *loop* tengah sehingga didapatkan persamaan sebagai berikut.

$$\ell_Q\left(\beta_0, \boldsymbol{\beta}\right) = -\frac{1}{2n}\sum_{i=1}^{n}\omega_i\left(z_i - \beta_0 - x_i^T\boldsymbol{\beta}\right)^2 + C\left(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}\right), \tag{2.46}$$

dengan respon dan bobot

$$z_i = \tilde{\beta}_0 + x_i^T\tilde{\boldsymbol{\beta}} + \frac{y_i - \tilde{P}(y_i = 1|x_i)}{\tilde{P}(y_i = 1|x_i)\tilde{P}(y_i = 0|x_i)}, \tag{2.47}$$

dan

$$\omega_i = \tilde{P}(y_i = 1|x_i)\tilde{P}(y_i = 0|x_i), \tag{2.48}$$

dimana $\tilde{P}(y_i = 1|x_i)$ dan $\tilde{P}(y_i = 0|x_i)$ dievaluasi pada perkiraan saat ini dan $C\left(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}\right)$ adalah konstan. Pada regresi logistik, *loss function* yang digunakan adalah *log-likelihood function* yaitu $\ell_Q\left(\beta_0, \boldsymbol{\beta}\right)$ yang merupakan kuadrat terkecil berboboti. Sementara itu algoritma penurunan koordinat digunakan untuk memecahkan masalah kuadrat terkecil tertimbang (PWLS) pada loop dalam adalah sebagai berikut.

$$\min_{\beta_0, \beta}\left\{-\ell_Q\left(\beta_0, \boldsymbol{\beta}\right) + \lambda R_\gamma\left(\boldsymbol{\beta}\right)\right\}. \tag{2.49}$$

Persamaan *regularized function* (2.49) bergantung pada bentuk *regularized* yang digunakan. Bentuk *regularized* untuk metode *lasso* adalah sebagai berikut.

$$\lambda R_\gamma\left(\boldsymbol{\beta}\right) = \lambda\left(\gamma\|\boldsymbol{\beta}\|_1\right), \tag{2.50}$$

dimana nilai γ = 1 dan λ > 0 adalah *tuning parameter*. Sementara bentuk *regularized* untuk metode *elastic-net* adalah sebagai berikut.

$$\lambda R_{\gamma}(\boldsymbol{\beta}) = \lambda\left[(1-\gamma)\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \gamma\|\boldsymbol{\beta}\|_1\right], \quad (2.51)$$

dengan dengan bobot $0 < \gamma < 1$ yang harus dioptimalkan bersamaan dengan $\lambda > 0$ yang merupakan *tuning parameter*.

Setiap *descent loop* koordinat dalam berlanjut sampai perubahan maksimum pada persamaan (2.49) kurang dari *threshold* yang sangat kecil dimana nilai *threshold* yang digunakan dalam penelitian ini adalah 1E-7. Langkah selanjutnya adalah mengurangi nilai λ dan mengulangi ketiga loop sampai mendapatkan estimasi yang konvergen. Proses yang dilakukan untuk optimalisasi γ dan λ dapat dilakukan dengan cara untuk γ yang nilainya tetap, *tuning parameter* λ dioptimalkan berdasarkan nilai AUC yang paling tinggi.

2.6 Ukuran Ketepatan Klasifikasi

Kemampuan prediksi dari algoritma klasifikasi biasanya diukur dengan akurasi prediksinya. Ketepatan klasifikasi digunakan untuk mengukur kebaikan dari model dalam memprediksi kelas berdasarkan kelas data aktual. *Area Under ROC Curve* (AUC) adalah ukuran ketepatan klasifikasi yang konsisten secara statistik dimana ukuran yang lebih baik daripada akurasi (Huang & Ling, 2005). Nilai AUC dihitung dengan menggunakan rata-rata perkiraan bidang berbentuk kurva yang dibentuk oleh TP_{rate} dan FP_{rate} (Dubey, et al., 2014). Nilai TP_{rate} dan FP_{rate} didapatkan dari *confusion matrix* yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 *Confussion Matrix Klasifikasi*

Aktual	Prediksi	
	Positif= kelas 0	Negatif = kelas 1
Positif= kelas 0	*True Positif* (TP)	*False Negative* (FN)
Negatif = kelas 1	*False Positif* (FP)	*True Negative* (TN)

dengan

TP : Jumlah anggota kelas 0 yang diprediksi dengan benar
FP : Jumlah anggota kelas 0 yang diprediksi dengan salah
FN : Jumlah anggota kelas 1 yang diprediksi dengan salah
TN : Jumlah anggota kelas 1 yang diprediksi dengan benar

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN}. \quad (2.52)$$

$$TP_{rate} = Sensitivity = \frac{TP}{TP + FN}. \quad (2.53)$$

$$Specificity = \frac{TN}{TN + FP}. \quad (2.54)$$

$$FP_{rate} = 1 - Specificity. \quad (2.55)$$

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2}. \quad (2.56)$$

Nilai maksimum AUC adalah sebesar 1 dimana dapat dikatakan model klasifikasi dapat memprediksi data dengan sempurna. Sementara itu apabila nilai AUC yang diperoleh adalah 0,5 menunjukkan bahwa model klasifikasi merupakan model acak tanpa kekuatan diskriminatif untuk memisahkan data (Härdle, et al., 2014).

2.7 Uji Beda *Mean*

Tujuan pengujian ini adalah untuk mengetahui perbedaan rata-rata dua kelompok data independen dimana data kelompok yang satu tidak tergantung dari data kelompok kedua. Hipotesis yang digunakan adalah sebagai berikut (Walpole, 2007).

$H_0: \mu_1 - \mu_2 = 0$

$H_0: \mu_1 - \mu_2 \neq 0$

dimana statistik uji yang digunakan adalah sebagai berikut.

$$z_{hit} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (2.57)$$

dengan,

$\bar{x}$: Rata-rata data sampel

σ : Standar deviasi data sampel

n : Banyaknya data sampel

Kriteria tolak H_0 apabila $|z_{hit}| > z_{\alpha/2}$.

(Halaman ini sengaja dikosongkan)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan pada penelitian ini adalah data sekunder. Data diperoleh dari data konsumen mengenai data pembayaran jasa pada Perusahaan Telco pada tahun 2018 (Natasha, 2019). Data pengamatan pada Bulan Januari sampai Juni 2019 merupakan pengamatan mengenai karakteristik pelanggan dalam enam bulan terakhir (M1 sampai M6) yang digunakan sebagai variabel prediktor. Sementara itu data pengamatan pada Bulan Juli hingga Desember 2018 merupakan data pengamatan apakah setiap pelanggan pernah atau tidak pernah memiliki keterlambatan pembayaran selama enam bulan yang digunakan sebagai variabel respon. Dataset dimana terdiri dari pengamatan terhadap 200.000 data pelanggan dengan perbandingan data variabel respon klasifikasi sebesar 1,002:1. Data dibagi menjadi 80% sebagai data *training* dan 20% sebagai data *testing*.

3.2 Variabel Penelitian

Variabel penelitian yang digunakan dalam penelitian ini ditunjukkan pada Tabel 3.1 berikut.

Tabel 3.1 Variabel Penelitian

Var	Kode	Deskripsi	Ket
y	Y_RISK	Pembayaran bulanan yang Terlambat dalam 6 Bulan Terakhir	Nominal 0: Tidak pernah telat bayar 1: Telat bayar minimal satu kali
x_1	GENDER	Jenis Kelamin	Nominal
x_2	SOCIAL_ECO_CAT	Kategori Sosial Ekonomi Konsumen	Ordinal

Tabel 3.1 Variabel Penelitian (Lanjutan)

Var	Kode	Deskripsi	Ket
x_3	TENURE	Lama Pemakaian Jasa (Bulan)	Rasio
x_4	VOICE_CALL_T_MIN6	Nilai Minimum Transaksi Telepon (*Outgoing* + *Incoming)* selama 6 bulan	Rasio
x_5	VOICE_CALL_T_MAX6	Nilai Maksimum Transaksi Telepon (*Outgoing* + *Incoming)* selama 6 bulan	Rasio
x_6	VOICE_CALL_T_SUM6	Total Transaksi Telepon (*Outgoing* + *Incoming)* selama 6 bulan	Rasio
x_7	VOICE_DUREE_T_MIN6	Nilai Minimum Durasi Telfon selama 6 bulan	Rasio
x_8	VOICE_DUREE_T_MAX6	Nilai Maksimum Durasi Telfon selama 6 bulan	Rasio
x_9	VOICE_DUREE_T_SUM6	Total Durasi Telfon selama 6 bulan	Rasio
x_{10}	INT_USAGE_MIN6	Nilai Minimum Penggunaan Internet (*Megabytes*/MB) selama 6 bulan	Rasio
x_{11}	INT_USAGE_MAX6	Nilai Maksimum Penggunaan Internet (*Megabytes*/MB) selama 6 bulan	Rasio

Tabel 3.1 Variabel Penelitian (Lanjutan)

Var	Kode	Deskripsi	Ket
x_{12}	INT_USAGE_SUM6	Total Penggunaan Internet (*Megabytes*/MB) selama 6 bulan	Rasio
x_{13}	ALL_TROUBLE_MIN6	Nilai Minimum Kejadian *Trouble* perbulan selama 6 bulan	Rasio
x_{14}	ALL_TROUBLE_MAX6	Nilai Maksimum Kejadian *Trouble* perbulan selama 6 bulan	Rasio
x_{15}	ALL_TROUBLE_SUM6	Total Kejadian *Trouble* perbulan selama 6 bulan	Rasio
x_{16}	ALL_MTTR_MIN6	Waktu Minimum Pembenahan Kejadian *Trouble*	Rasio
x_{17}	ALL_MTTR_MAX6	Waktu Maksimum Pembenahan Kejadian *Trouble*	Rasio
x_{18}	ALL_MTTR_SUM6	Total Minimum Pembenahan Kejadian *Trouble*	Rasio
x_{19}	VOICE_PAYMENT_MIN6	Minimum Pembayaran Telefon	Rasio
x_{20}	VOICE_PAYMENT_MAX6	Maksimum Pembayaran Telefon	Rasio
x_{21}	VOICE_PAYMENT_SUM6	Total Pembayaran Telefon	Rasio
x_{22}	VOICE_PY_RATIO_M1	Rasio Pembayaran Telefon Bulan 1 terhadap Rerata Total Pembayaran	Rasio

Tabel 3.1 Variabel Penelitian (Lanjutan)

Var	Kode	Deskripsi	Ket
x_{23}	VOICE_PY_RATIO_M2	Rasio Pembayaran Telefon Bulan 2 terhadap Rerata Total Pembayaran	Rasio
x_{24}	VOICE_PY_RATIO_M3	Rasio Pembayaran Telefon Bulan 3 terhadap Rerata Total Pembayaran	Rasio
x_{25}	VOICE_PY_RATIO_M4	Rasio Pembayaran Telefon Bulan 4 terhadap Rerata Total Pembayaran	Rasio
x_{26}	VOICE_PY_RATIO_M5	Rasio Pembayaran Telefon Bulan 5 terhadap Rerata Total Pembayaran	Rasio
x_{27}	VOICE_PY_RATIO_M6	Rasio Pembayaran Telefon Bulan 6 terhadap Rerata Total Pembayaran	Rasio
x_{28}	INT_PAYMENT_MIN6	Minimum Pembayaran Internet	Rasio
x_{29}	INT_PAYMENT_MAX6	Maksimum Pembayaran Internet	Rasio
x_{30}	INT_PAYMENT_SUM6	Total Pembayaran Internet	Rasio
x_{31}	INT_PY_RATIO_M1	Rasio Pembayaran Internet Bulan ke-1 terhadap Rerata Total Pembayaran	Rasio
x_{32}	INT _PY_RATIO_M2	Rasio Pembayaran Internet Bulan ke-2 terhadap Rerata Total Pembayaran	Rasio

Tabel 3.1 Variabel Penelitian (Lanjutan)

Var	Kode	Deskripsi	Ket
x_{33}	INT _PY_RATIO_M3	Rasio Pembayaran Internet Bulan ke-3 terhadap Rerata Total Pembayaran	Rasio
x_{34}	INT _PY_RATIO_M4	Rasio Pembayaran Internet Bulan ke-4 terhadap Rerata Total Pembayaran	Rasio
x_{35}	INT _PY_RATIO_M5	Rasio Pembayaran Internet Bulan ke-5 terhadap Rerata Total Pembayaran	Rasio
x_{36}	INT _ PY_RATIO_M6	Rasio Pembayaran Internet Bulan ke-6 terhadap Rerata Total Pembayaran	Rasio

Variabel gender pada kategori 0 merupakan pelanggan perempuan dan 1 merupakan pelanggan laki-laki. Kategori social ekonomi pelanggan dibagi menjadi 7 kelas. Kelas pertama merupakan kelas pedesaan bawah, kedua merupakan pedesaan menengah dan ketiga merupakan kelas pedesaan atas. Kelas keempat adalah kelas perkotaan bawah, kelima merupakan perkotaan tengah, kelas keenam merupakan kelas perkotaan elit dan kelas ketujuh merupakan kelas perkotaan kosmopolitan.

3.3 Struktur Data

Struktur data secara umum yang digunakan dalam penelitian ini disajikan pada Tabel 3.2 berikut.

Tabel 3.2 Struktur Data Penelitian

n	x_1	x_2	x_3	…	x_{36}	y
1	$x_{1,1}$	$x_{2,1}$	$x_{3,1}$	…	$x_{36,1}$	y_1
2	$x_{1,2}$	$x_{2,2}$	$x_{3,2}$	…	$x_{36,2}$	y_2
3	$x_{1,3}$	$x_{2,3}$	$x_{3,3}$	…	$x_{36,3}$	y_3
⋮	⋮	⋮	⋮	⋮	⋮	⋮
200000	$x_{1,200000}$	$x_{2,200000}$	$x_{3,200000}$	…	$x_{36,200000}$	y_{200000}

3.4 Langkah Penelitian

Langkah analisis yang digunakan dalam penelitian ini antara lain adalah sebagai berikut.

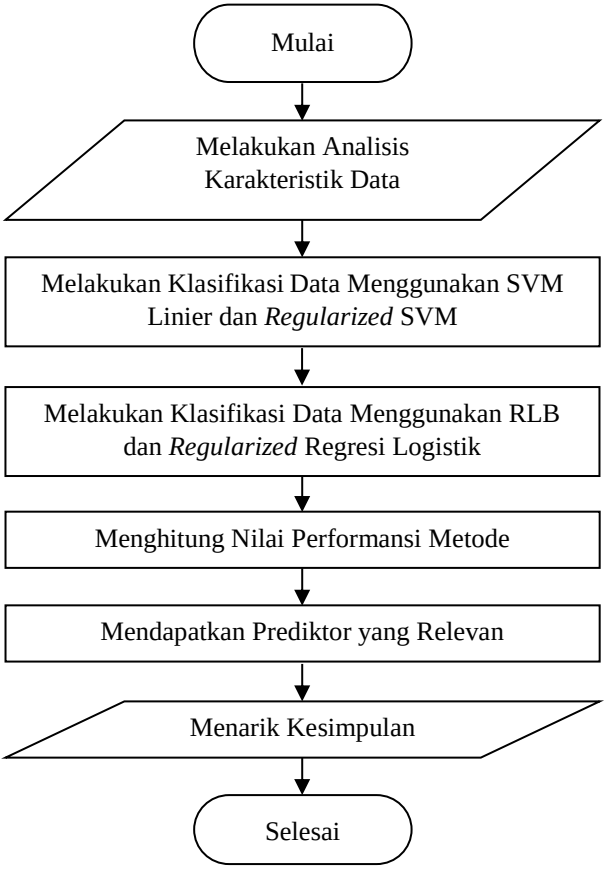
1. Melakukan analisis terhadap karakteristik data
   a. Membuat histogram untuk data kategorik.
   b. Membuat boxplot untuk data kontinyu.
   c. Melakukan uji beda *mean* pada data kontinyu.
2. Memprediksi telat bayar menggunakan metode linier SVM dan *regularized* SVM.
   a. Linier SVM
   i. Melakukan *fitting* model SVM linier.
   ii. Melakukan prediksi terhadap data *testing* berdasarkan model yan didapatkan.
   iii. Menghitung performansi metode pada data *training* dan *testing* berdasarkan nilai akurasi dan AUC.
   b. *Lasso* SVM
   i. Melakukan tuning parameter dengan menentukan nilai-nilai λ yang akan digunakan.
   ii. Melakukan *fitting* model SVM dengan menggunakan penalty *l*1.
   iii. Melakukan prediksi terhadap data *testing* berdasarkan model yan didapatkan.
   iv. Menghitung performansi metode pada data *training* dan *testing* berdasarkan nilai akurasi dan AUC.
   v. Mendapatkan nilai λ optimal dan subset variabel yang relevan berdasarkan nilai performansi metode yang paling tinggi.
   c. *Elastic-Net* SVM
   i. Melakukan tuning parameter dengan menentukan nilai-nilai λ_1 dan λ_2 yang akan digunakan.
   ii. Melakukan *fitting* model SVM dengan menggunakan penalty *elastic net*.
   iii. Melakukan prediksi terhadap data *testing* berdasarkan model yan didapatkan.
   iv. Menghitung performansi metode pada data *training* dan *testing* berdasarkan nilai akurasi dan AUC.

v. Mendapatkan nilai λ_1 dan λ_2 optimal dan subset variabel yang relevan berdasarkan nilai performansi metode yang paling tinggi.

3. Memprediksi telat bayar menggunakan metode regresi logistik dan *regularized* regresi logistik.
   a. Regresi logistik biner
   i. Melakukan *fitting* model regresi logistik.
   ii. Melakukan prediksi terhadap data *testing* berdasarkan model yan didapatkan.
   iii. Menghitung performansi metode pada data *training* dan *testing* berdasarkan nilai akurasi dan AUC.
   b. *Lasso* regresi logistik
   i. Menetapkan nilai $\gamma = 1$
   ii. Melakukan tuning parameter dengan menentukan nilai-nilai λ yang akan digunakan.
   iii. Melakukan *fitting* model dengan menggunakan penalty *l*1.
   iv. Melakukan prediksi terhadap data *testing* berdasarkan model yan didapatkan.
   v. Menghitung performansi metode pada data *training* dan *testing* berdasarkan nilai akurasi dan AUC.
   vi. Mendapatkan nilai λ optimal dan subset variabel yang relevan berdasarkan nilai performansi metode yang paling tinggi.
   c. *Elastic-net* regresi logistik
   i. Melakukan tuning parameter dengan menentukan nilai-nilai λ dan γ yang akan digunakan.
   ii. Melakukan *fitting* model dengan menggunakan penalty *elastic net*.
   iii. Melakukan prediksi terhadap data *testing* berdasarkan model yan didapatkan.
   iv. Menghitung performansi metode pada data *training* dan *testing* berdasarkan nilai akurasi dan AUC.
   v. Mendapatkan nilai λ dan γ optimal dan subset variabel yang relevan berdasarkan nilai performansi metode yang paling tinggi.

4. Mendapatkan subset variabel yang relevan
   a. Membandingkan hasil nilai akurasi dan nilai AUC pada masing-masing metode.
   b. Mendapatkan prediktor yang relevan berdasarkan metode terbaik.

Berdasarkan langkah penelitian yang telah dijelaskan dapat digambarkan diagram alir penelitian ini yang disajikan pada Gambar 3.1.



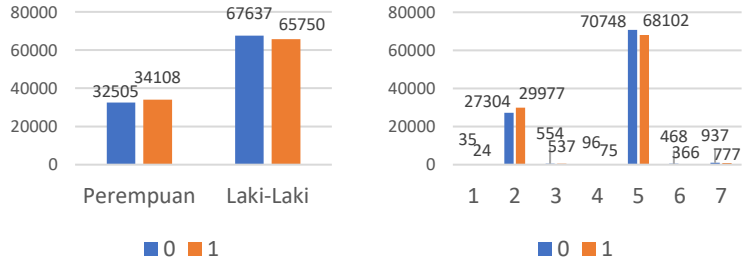
Gambar 3.1 Diagram Alir Penelitian

BAB IV
ANALISIS DAN PEMBAHASAN

Analisis dan pembahasan menyajikan hasil dari output dari proses yang telah dilakukan, dimana output ini menjawab tujuan penelitian. Pembahasan yang terdapat pada penelitian ini yaitu mengenai karakteristik data, prediksi menggunakan metode SVM dan regresi logistik dan hasil analisis mengenai prediktor yang relevan.

4.1 Karakteristik Data

Ekplorasi data bertujuan untuk melihat karakteristik dari data tersebut atau mendapatkan gambaran umum sebagai informasi awal dari sebuah data sebelum menentukan atau menerapkan metode analisis yang tepat. Karakteristik suatu data dapat diketahui melalui statistika deskriptif data tersebut. Statistika deskriptif data telat bayar dapat divisualisasikan pada Gambar 4.1 berikut.

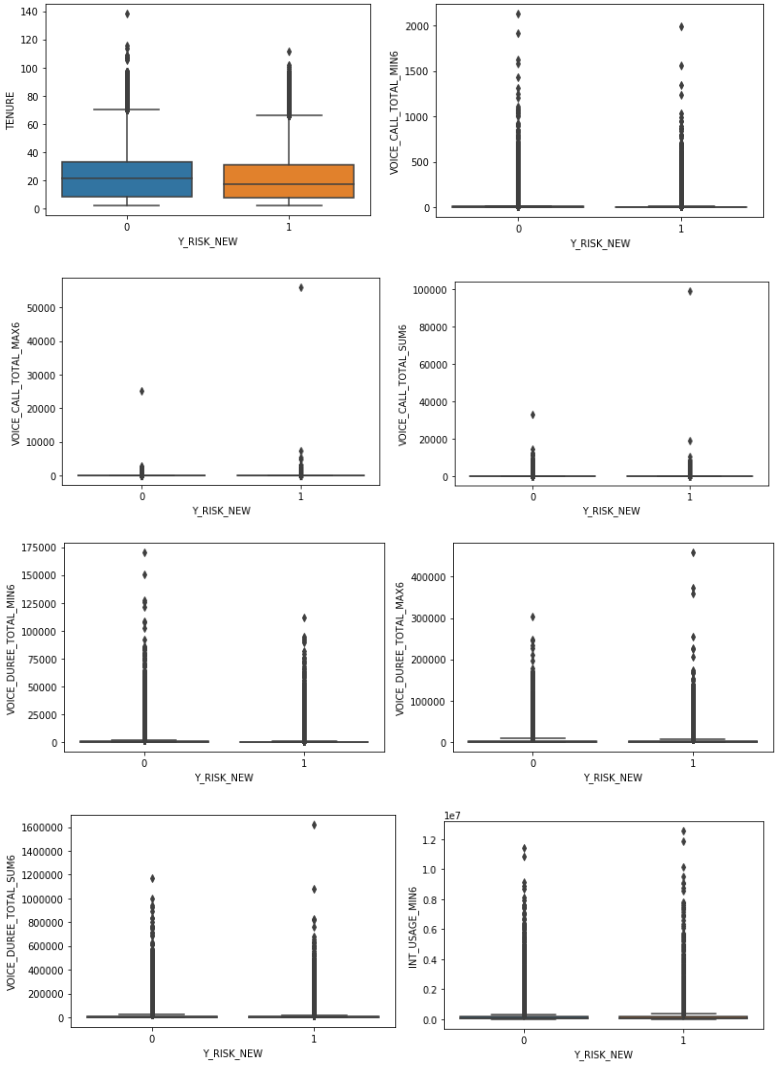


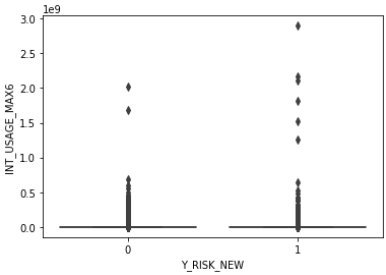
(a) (b)

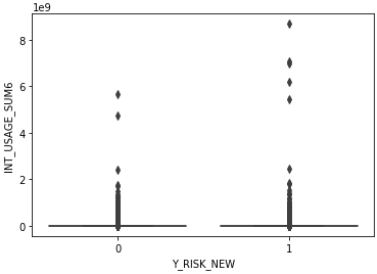
Gambar 4.1 Histogram Data Variabel Kategorik (a) Gender dan (b) Sosial Ekonomi

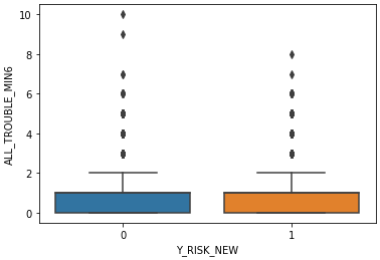
Gambar 4.1 menunjukkan bahwa pada variabel gender, pelanggan yang paling banyak adalah berjenis kelamin laki-laki. Jumlah pelanggan perempuan yang telat membayar dan tidak telat membayar hampir sama, begitu pula dengan pelanggan laki-laki. Hal ini juga terjadi pada variabel kategori sosial ekonomi. Jumlah pelanggan yang telat membayar dan tidak telat membayar memiliki nilai yang tidak berbeda jauh. Sementara kategori sosial ekonomi

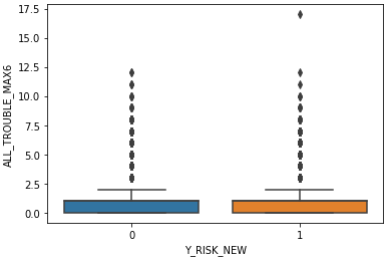
pelanggan yang paling banyak adalah kategori 5 yaitu kelas perkotaan menengah.

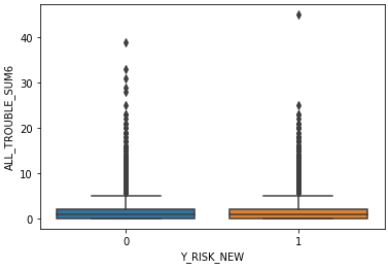


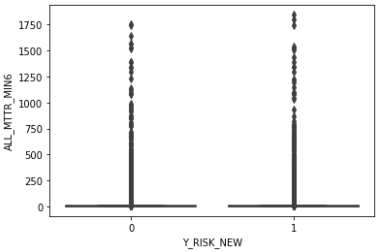

1e9
INT_USAGE_MAX6
Y_RISK_NEW

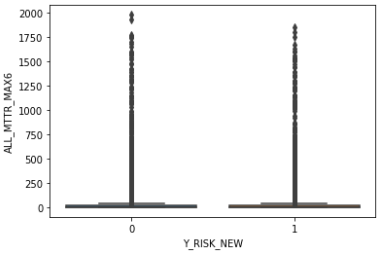

1e9
INT_USAGE_SUM6
Y_RISK_NEW

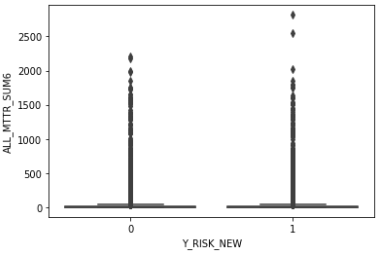

ALL_TROUBLE_MIN6
Y_RISK_NEW

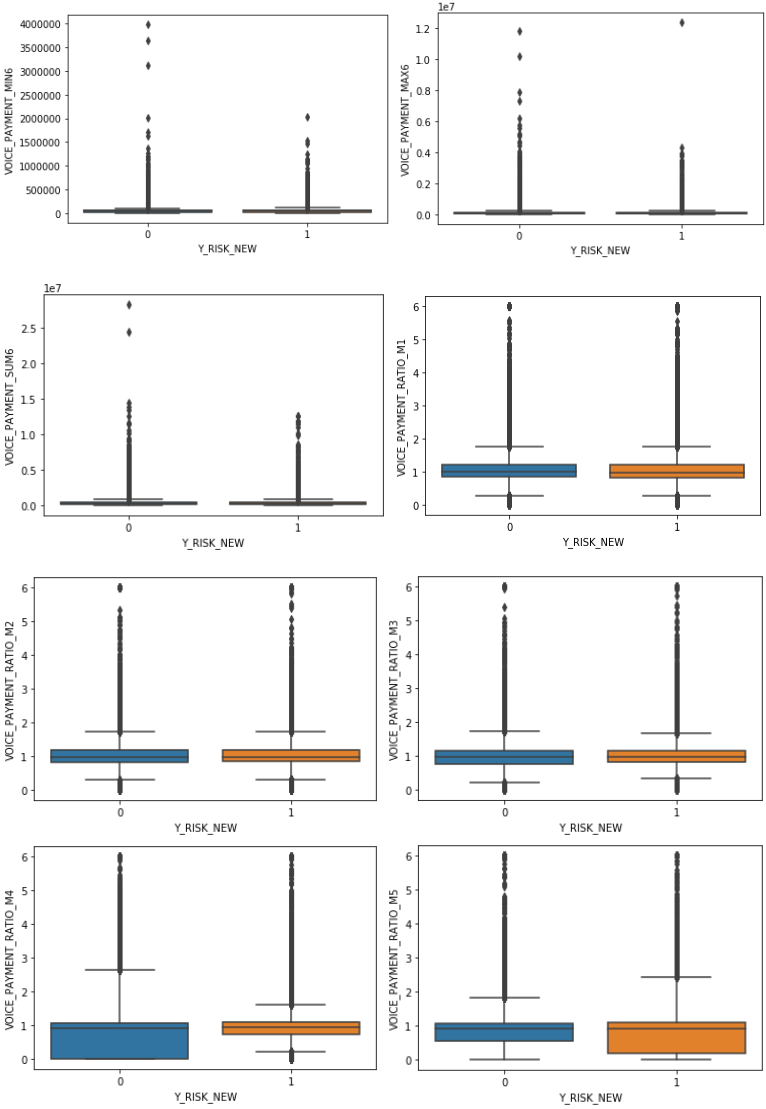

ALL_TROUBLE_MAX6
Y_RISK_NEW

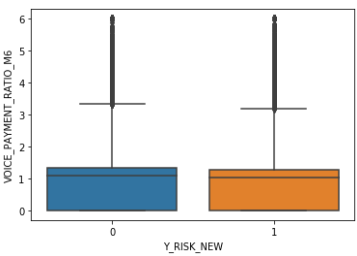

ALL_TROUBLE_SUM6
Y_RISK_NEW

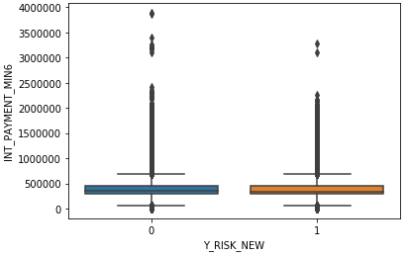

ALL_MTTR_MIN6
Y_RISK_NEW

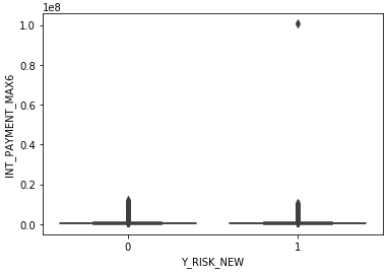

ALL_MTTR_MAX6
Y_RISK_NEW

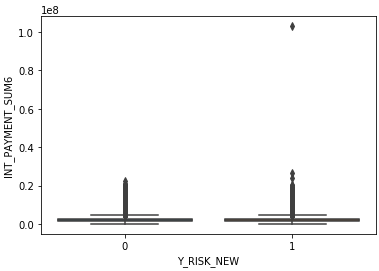

ALL_MTTR_SUM6
Y_RISK_NEW

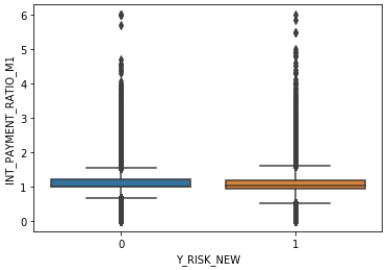

VOICE_PAYMENT_MIN6
VOICE_PAYMENT_MAX6
VOICE_PAYMENT_SUM6
VOICE_PAYMENT_RATIO_M1
VOICE_PAYMENT_RATIO_M2
VOICE_PAYMENT_RATIO_M3
VOICE_PAYMENT_RATIO_M4
VOICE_PAYMENT_RATIO_M5
Y_RISK_NEW
1e7

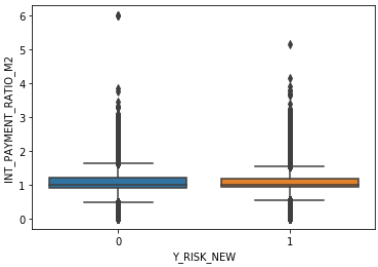

VOICE_PAYMENT_RATIO_M6
Y_RISK_NEW

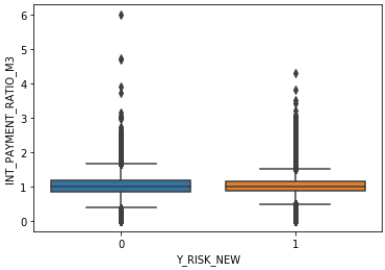

INT_PAYMENT_MIN6
Y_RISK_NEW

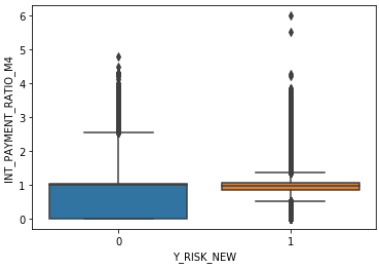

1e8
INT_PAYMENT_MAX6
Y_RISK_NEW

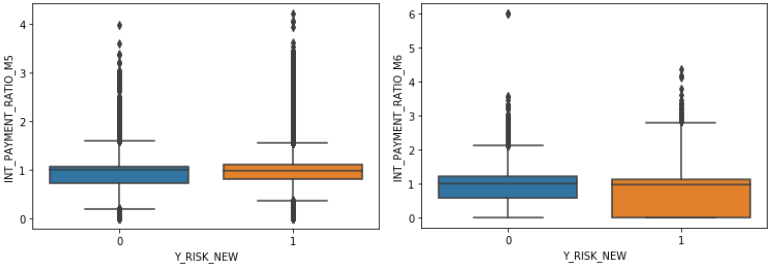

1e8
INT_PAYMENT_SUM6
Y_RISK_NEW


INT_PAYMENT_RATIO_M1
Y_RISK_NEW


INT_PAYMENT_RATIO_M2
Y_RISK_NEW


INT_PAYMENT_RATIO_M3
Y_RISK_NEW


INT_PAYMENT_RATIO_M4
Y_RISK_NEW



Gambar 4.2 Boxplot Data Variabel Kontinyu

Gambar 4.2 menunjukkan boxplot dari 34 variabel kontinyu. Setiap variabel memiliki *outlier* yang sangat banyak. Berdasarkan variabel responnya yaitu resiko keterlambatan pembayaran, setiap variabel kontinyu memiliki nilai *mean* yang sama antara pelanggan yang telat membayar dan tidak telat membayar. Hal ini dapat diartikan bahwa variabel prediktor kontinyu tidak dapat membedakan antara pelanggan yang telat membayar dan tidak telat membayar.

Tabel 4.1 Uji Beda *Mean*

Variabel	**Z_{hit}**	**Keputusan**
Tenure	28.836	Tolak H0
Voice_Call_Total_Min6	19.566	Tolak H0
Voice_Call_Total_Max6	6.9314	Tolak H0
Voice_Call_Total_Sum6	14.554	Tolak H0
Voice_Duree_Total_Min6	22.643	Tolak H0
Voice_Duree_Total_Max6	21.732	Tolak H0
Voice_Duree_Total_Sum6	24.038	Tolak H0
Int_Usage_Min6	1.080	Gagal Tolak H0
Int_Usage_Max6	-1.599	Gagal Tolak H0
Int_Usage_Sum6	-1.814	Gagal Tolak H0
All_Trouble_Min6	-3.949	Tolak H0
All_Trouble_Max6	-1.897	Gagal Tolak H0
All_Trouble_Sum6	-2.872	Tolak H0
All_Mttr_Min6	-0.202	Gagal Tolak H0
All_Mttr_Max6	-1.272	Gagal Tolak H0

Tabel 4.1 Uji Beda *Mean* (lanjutan)

Variabel	Z_{hit}	Keputusan
All_Mttr_Sum6	-1.774	Gagal Tolak H0
Voice_Payment_Min6	9.416	Tolak H0
Voice_Payment_Max6	16.750	Tolak H0
Voice_Payment_Sum6	6.386	Tolak H0
Voice_Payment_Ratio_M1	22.016	Tolak H0
Voice_Payment_Ratio_M2	-13.531	Tolak H0
Voice_Payment_Ratio_M3	-20.731	Tolak H0
Voice_Payment_Ratio_M4	-12.597	Tolak H0
Voice_Payment_Ratio_M5	-9.827	Tolak H0
Voice_Payment_Ratio_M6	16.617	Tolak H0
Int_Payment_Min6	13.827	Tolak H0
Int_Payment_Max6	20.080	Tolak H0
Int_Payment_Sum6	-1.703	Gagal Tolak H0
Int_Payment_Ratio_M1	31.514	Tolak H0
Int_Payment_Ratio_M2	-15.550	Tolak H0
Int_Payment_Ratio_M3	-28.061	Tolak H0
Int_Payment_Ratio_M4	-12.893	Tolak H0
Int_Payment_Ratio_M5	-16.475	Tolak H0
Int_Payment_Ratio_M6	18.414	Tolak H0

Uji beda *mean* dilakukan untuk mengetahui perbedaan rata-rata yang signifikan pada setiap variabel kontinyu dimana hal ini belum bisa dijelaskan dari visualisasi data menggunakan *boxplot*. Nilai z_{hit} dibandingkan dengan z_{tabel} yaitu 1,96. Tabel 4.1 menunjukkan bahwa ada delapan variabel kontinyu yang memiliki rata-rata yang sama berdasarkan variabel respon yaitu pelanggan yang terlambat dan yang tidak terlambat. Variabel yang tidak memiliki perbedaan rata-rata adalah variabel penggunaan internet minimam, maksimal dan total selama enam bulan, maksimal terjadinya *trouble*, waktu minimum, maksimum dan total perbaikan, serta total pembayaran internet selama enam bulan.

4.2 Klasifikasi Menggunakan *Support Vector Machines* (SVM)

Data yang digunakan untuk analisis SVM terdiri dari 10 data *training* dan 10 data *testing*, dimana data tersebut didapatkan menggunakan *stratified random sampling*. *Standardize* dilakukan sebelum data dianalisis. Hal ini dikarenakan data memiliki skala yang berbeda jauh. Prediksi menggunakan metode SVM dilakukan dengan dua metode berbeda yaitu SVM linier dan *regularized* SVM. *Regularized* SVM yang digunakan pada penelitian ini yaitu dengan menggunakan penalti *lasso* dan *elastic-net.*

4.2.1 SVM Linier

SVM linier merupakan metode yang memisahkan data secara linier. Metode ini menghasilkan prediksi data tanpa adanya *feature selection*. Hasil kebaikan model pada setiap skenario data disajikan pada Tabel 4.2 berikut.

Tabel 4.2 Nilai Kebaikan Model SVM Linier

Data	Training		Testing	
	Akurasi	**AUC**	**Akurasi**	**AUC**
1	0.58594	0.58602	0.59042	0.59049
2	0.58696	0.58704	0.59005	0.59013
3	0.58742	0.58750	0.58285	0.58293
4	0.58713	0.58721	0.58637	0.58645
5	0.58697	0.58705	0.58647	0.58655
6	0.58724	0.58732	0.58755	0.58762
7	0.58734	0.58743	0.58822	0.58831
8	0.58673	0.58681	0.58892	0.58899
9	0.58696	0.58704	0.58722	0.58730
10	0.58756	0.58765	0.58857	0.58865

Tabel 4.2 menunjukkan bahwa data ke-1 memiliki nilai AUC *testing* paling tinggi yaitu 0,59049. Sementara nilai AUC *training* paling tinggi adalah 0,58765 pada data ke-10. Model SVM linier yang didapatkan tidak menunjukkan adanya *overfitting*. Rata-rata

nilai akurasi pada data *training* yaitu sebesar 0,587025 dan pada data *testing* sebesar 0,58774. Nilai akurasi pada data *testing* menunjukkan hasil yang lebih tinggi daripada data *training*, namun perbedaannya tidak terlalu besar. Nilai kebaikan model yang didapatkan menunjukkan bahwa SVM linier belum mampu secara baik mengklasifikasikan data telat bayar.

4.2.2 *Lasso* SVM

Nilai parameter yang dicobakan pada metode *lasso* SVM adalah *C*=[2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 7]. Parameter *C* merupakan nilai yang berbanding terbalik dengan *regularized parameter* yaitu λ. Semakin besar nilai *C* maka nilai λ akan semakin kecil. Nilai λ yang mendekati 0 akan membuat variabel terpilih semakin banyak, begitu pula sebaliknya jika nilai λ mendekati tak hingga maka variabel yang terpilih akan semakin sedikit. Pemilihan λ optimal didasarkan pada hasil AUC data *testing* dan didapatkan hasil terbaik pada masing-masing data pada Tabel 4.3.

Tabel 4.3 Nilai Kebaikan Model *Lasso* SVM

Data	C	λ	p*	Training		Testing	
				Akurasi	AUC	Akurasi	AUC
1	2.5	0.400	35	0.59233	0.59241	0.59505	0.59512
2	4.5	0.222	35	0.59358	0.59366	0.59568	0.59575
3	7.0	0.142	36	0.59364	0.59372	0.58835	0.58844
4	3.0	0.333	36	0.59281	0.59289	0.59333	0.59340
5	5.0	0.200	36	0.59302	0.59310	0.59198	0.59205
6	6.0	0.167	35	0.59313	0.59321	0.59290	0.59298
7	2.5	0.400	34	0.59354	0.59362	0.59240	0.59249
8	7.0	0.142	35	0.59278	0.59286	0.59240	0.59247
9	3.0	0.333	34	0.59326	0.59334	0.59308	0.59316
10	3.0	0.333	34	0.59378	0.59386	0.59315	0.59323

Model terbaik berdasarkan Tabel 4.3 adalah model pada data ke-2. Model ini memiliki nilai AUC dan akurasi paling tinggi. Nilai parameter λ optimal pada model ini adalah 0,22 dimana ada 35

variabel terpilih dari total 36 variabel yang ada. Nilai akurasi dan AUC baik pada data *training* maupun data *testing* tidak berbeda jauh. Hal ini menandakan bahwa sudah tidak terjadi *overfitting*. Nilai AUC yang berkisar diantara 0,59 menandakan metode yang digunakan belum mampu mengklasifikasikan data telat bayar dengan baik. Salah satu penyebabnya adalah boxplot variabel kontinyu berdasarkan variabel responnya dan histogram variabel kategorik berdasarkan variabel responnya tidak signifikan. Hal ini mengartikan bahwa tidak ada perbedaan antara data antara kategori terlambat bayar dan tidak pernah terlambat bayar.

4.2.3 *Elastic-Net* SVM

Nilai parameter yang dicobakan pada metode ini adalah kombinasi antara λ_1=[0.0001, 0.0003, 0.0005, 0.0007, 0.0009] dan λ_2=[0.2, 0.4, 0.6, 0.8]. Kedua nilai parameter ini dicobakan pada seluruh data dan diperoleh 20 model kombinasi dimana model terbaik terdapat pada Tabel 4.4 berikut. Apabila nilai λ_2 konstan, maka semakin besar nilai λ_1 maka variabel yang terpilih akan semakin sedikit variabel yang terpilih. Sementara apabila λ_1 bernilai konstan semakin besar nilai λ_2 maka variabel yang terpilih akan semakin sedikit variabel yang terpilih.

Tabel 4.4 Nilai Kebaikan Model *Elastic-Net* SVM

Data	λ_1	λ_2	p*	Training		Testing	
				Akurasi	AUC	Akurasi	AUC
1	0.0007	0.2	24	0.55658	0.55665	0.55813	0.55820
2	0.0007	0.8	17	0.55498	0.55508	0.55955	0.55964
3	0.0007	0.6	21	0.55650	0.55661	0.55210	0.55222
4	0.0003	0.8	18	0.55861	0.55870	0.55863	0.55871
5	0.0009	0.2	22	0.55624	0.55631	0.55735	0.55742
6	0.0005	0.2	24	0.55911	0.55911	0.56345	0.56345
7	0.0007	0.4	23	0.55896	0.55894	0.55840	0.55839
8	0.0001	0.2	32	0.56120	0.56129	0.56403	0.56411
9	0.0003	0.8	19	0.56159	0.56163	0.55958	0.55962
10	0.0007	0.2	25	0.55763	0.55768	0.55730	0.55735

Nilai AUC data *testing* pada Tabel 4.4 menunjukkan bahwa data ke-8 merupakan model terbaik karena memiliki nilai AUC paling tinggi. Pada model ini terpilih sebanyak 32 variabel dengan nilai λ_1 sebesar 0,0003 dan λ_2 sebesar 0,2. Sama halnya seperti metode *lasso* SVM, metode *elastic-net* juga belum mampu menghasilkan nilai AUC yang tinggi. Hasil kebaikan model pada data *training* dan *testing* juga menunjukkan bahwa tidak terjadi *overfitting*.

4.3 Klasifikasi Menggunakan Regresi Logistik

Klasifikasi menggunakan metode regresi logistik digunakan pada 10 macam data *training* dan *testing* sama halnya dengan metode *regularized* SVM. Metode regresi logistik yang dibahas pada penelitian ini adalah penggunaan metode regresi logistik biner dan juga *regularized* regresi logistik (*lasso* dan *elastic-net*). Data yang digunakan dalam analisis merupakan data yang telah dilakukan *standardize*.

4.3.1 Regresi Logistik Biner

Metode regresi logistik biner diterapkan untuk memprediksi pelanggan Perusahaan Telco akan terlambat membayar tagihan atau tidak. Hasil klasifikasi telat bayar menggunakan regresi logistik pada 10 macam skenario data disajikan pada Tabel 4.5. Hasil kebaikan model didasarkan pada nilai akurasi dan AUC pada data *training* dan data *testing*.

Tabel 4.5 Nilai Kebaikan Model Regresi Logistik

Data	Training		Testing	
	Akurasi	**AUC**	**Akurasi**	**AUC**
1	0.57200	0.57206	0.57527	0.57531
2	0.57271	0.57276	0.57492	0.57498
3	0.57303	0.57309	0.57142	0.57148
4	0.57365	0.57370	0.57357	0.57362
5	0.57360	0.57366	0.57107	0.57113
6	0.57316	0.57321	0.57472	0.57397
7	0.57346	0.57352	0.57222	0.57228

Tabel 4.5 Nilai Kebaikan Model Regresi Logistik (Lanjutan)

Data	Training		Testing	
	Akurasi	**AUC**	**Akurasi**	**AUC**
8	0.57274	0.57279	0.57565	0.57570
9	0.57303	0.57309	0.57375	0.57380
10	0.57300	0.57305	0.57450	0.57455

Tabel 4.5 menunjukkan data ke-8 mempunyai nilai AUC pada data *testing* paling tinggi sehingga dapat dikatakan skenario pada data ke-8 merupakan yang paling baik diantara yang lainnya. Sementara pada data *training*, data ke-4 memiliki nilai AUC paling tinggi. Selisih pada nilai akurasi dan nilai AUC pada masing-masing skenario data tidak terlalu jauh. Rata-rata nilai akurasi pada data *training* dan *testing* adalah sebesar 0,57303 dan 0,57370. Hasil estimasi parameter pada model terbaik ditunjukkan pada Tabel 4.6 berikut.

Tabel 4.6 Estimasi Parameter Regresi Logistik Biner

Parameter	**Koefisien**	***P-Value***
Constant	1.368	0.000*
Gender	-0.066	0.000*
Social_Eco_Cat_2	-0.101	0.152
Social_Eco_Cat_3	-0.276	0.000*
Social_Eco_Cat_4	-0.367	0.000*
Social_Eco_Cat_5	-0.381	0.034*
Social_Eco_Cat_6	-0.121	0.000*
Social_Eco_Cat_7	-0.411	0.171
Tenure	-1.198	0.000*
Voice_Call_Total_Min6	-4.820	0.003*
Voice_Call_Total_Max6	-9.181	0.399
Voice_Call_Total_Sum6	18.384	0.160
Voice_Duree_Total_Min6	4.466	0.000*
Voice_Duree_Total_Max6	6.257	0.000*
Voice_Duree_Total_Sum6	-16.825	0.000*

Tabel 4.6 Estimasi Parameter Regresi Logistik Biner (Lanjutan)

Parameter	Koefisien	*P-Value*
Int_Usage_Min6	-0.193	0.566
Int_Usage_Max6	-8.468	0.294
Int_Usage_Sum6	9.897	0.218
All_Trouble_Min6	0.857	0.000*
All_Trouble_Max6	-1.209	0.000*
All_Trouble_Sum6	1.137	0.003*
All_Mttr_Min6	-0.128	0.770
All_Mttr_Max6	-1.409	0.264
All_Mttr_Sum6	1.867	0.243
Voice_Payment_Min6	-1.642	0.143
Voice_Payment_Max6	-10.200	0.000*
Voice_Payment_Sum6	14.116	0.000*
Voice_Payment_Ratio_M1	-0.648	0.000*
Voice_Payment_Ratio_M2	-0.031	0.782
Voice_Payment_Ratio_M3	-0.252	0.025*
Voice_Payment_Ratio_M4	0.059	0.573
Voice_Payment_Ratio_M5	-0.233	0.012*
Voice_Payment_Ratio_M6	-0.230	0.001*
Int_Payment_Min6	-9.772	0.000*
Int_Payment_Max6	-110.155	0.000*
Int_Payment_Sum6	64.532	0.000*
Int_Payment_Ratio_M1	-4.100	0.000*
Int_Payment_Ratio_M2	0.870	0.000*
Int_Payment_Ratio_M3	0.743	0.000*
Int_Payment_Ratio_M4	-0.426	0.001*
Int_Payment_Ratio_M5	0.115	0.142
Int_Payment_Ratio_M6	-1.962	0.000*

Hasil pengujian serentak didapatkan *p-value* pada *likelihood ratio test* sebesar 0 dimana dapat disimpulkan bahwa minimal terdapat satu parameter yang signifikan terhadap model. Namun

pada uji parsial pada Tabel 4.6 menunjukkan bahwa terdapat 14 variabel yang tidak signifikan diantaranya adalah jumah total transaksi telepon, waktu minimum dan maksimum pembenahan *trouble*. Kemudian dilakukan seleksi variabel menggunakan metode *backward* dimana menghasilkan nilai estimasi sebagai berikut.

Tabel 4.7 Estimasi Parameter *Backward* Regresi Logistik Biner

Parameter	**Koefisien**	***P-Value***
Constant	1.397	0.000
Gender	-0.066	0.000
Social_Eco_Cat_2	-0.101	0.149
Social_Eco_Cat_3	-0.275	0.000
Social_Eco_Cat_4	-0.366	0.000
Social_Eco_Cat_5	-0.382	0.034
Social_Eco_Cat_6	-0.121	0.000
Social_Eco_Cat_7	-0.409	0.172
Tenure	-1.197	0.000
Voice_Call_Total_Min6	-2.386	0.000
Voice_Duree_Total_Min6	3.002	0.001
Voice_Duree_Total_Max6	5.974	0.000
Voice_Duree_Total_Sum6	-14.902	0.000
All_Trouble_Min6	0.846	0.000
All_Trouble_Max6	-1.296	0.000
All_Trouble_Sum6	1.348	0.000
Voice_Payment_Max6	-9.180	0.000
Voice_Payment_Sum6	12.162	0.000
Voice_Payment_Ratio_M1	-0.669	0.000
Voice_Payment_Ratio_M3	-0.238	0.027
Voice_Payment_Ratio_M5	-0.133	0.036
Voice_Payment_Ratio_M6	-0.253	0.000
Int_Payment_Min6	-9.936	0.000
Int_Payment_Max6	-111.910	0.000

Tabel 4.7 Estimasi Parameter *Backward* Regresi Logistik Biner (Lanjutan)

Parameter	Koefisien	*P-Value*
Int_Payment_Sum6	65.669	0.000
Int_Payment_Ratio_M1	-4.143	0.000
Int_Payment_Ratio_M2	0.796	0.000
Int_Payment_Ratio_M3	0.715	0.000
Int_Payment_Ratio_M4	-0.388	0.000
Int_Payment_Ratio_M6	-1.954	0.000

Tabel 4.7 menunjukkan variabel-variabel yang signifikan mempengaruhi telat bayar pelanggan berdasarkan metode *backward* regresi logistik. Terdapat 24 variabel yang signifikan terhadap model regresi losgistik diantaranya adalah, *gender*, *social economy category*, *tenure* dan yang lainnya. Model regresi logistik yang didapatkan adalah sebagai berikut.

$$g(x) = 1,397 - 0,066\ \text{Gender} + ... - 1,955\ \text{Int\_Payment\_Ratio\_M6}.$$

Nilai *odd ratio* pada model regresi logistik biner dapat dihitung berdasarkan nilai eksponen dari koefisien parameternya. Pelanggan perempuan cenderung akan telat membayar 1,06 kali dari pada pelanggan laki-laki. Pelanggan dari kelas pedesaan bawah akan cenderung 1,31 kali telat membayar tagihan daripada pelanggan kelas pedesaan atas. Pelanggan dari kelas sosial ekonomi pertama (pedesaan bawah) cenderung akan 1,46 kali dibandingkan dengan pelanggan kelas kelima (perkotaan menengah). Apabila pelanggan menambah masa penggunaan jasa selama 6 bulan maka akan cenderung memiliki 1,49 kali untuk tidak telat membayar. Setelah didapatkan model, selanjutnya dilakukan prediksi dengan model terbaik. Prediksi menggunakan metode *backward* didapatkan nilai akurasi pada data *training* dan *testing* adalah 0,57081 dan 0,57572. Sementara nilai AUC data *training* dan data *testing* yang didapatkan adalah sebesar 0,57087 dan 0,57578.

4.3.2 *Lasso* Regresi Logistik

Nilai parameter *C* yang dicobakan pada metode ini adalah 0,5 sampai 5 dengan kelipatan 0,5. Parameter *C* merupakan nilai

yang berbanding terbalik dengan *regularized parameter* yaitu λ. Hasil model terbaik pada masing-masing skenario data disajikan pada Tabel 4.8 yang diukur berdasarkan nilai akurasi dan nilai AUC pada data *training* dan data testing.

Tabel 4.8 Nilai Kebaikan Model *Lasso* Regresi Logistik

Data	C	λ	p*	Training		Testing	
				Akurasi	AUC	Akurasi	AUC
1	3.0	0.333	33	0.59307	0.59315	0.59633	0.59640
2	3.5	0.285	34	0.59354	0.59362	0.59600	0.59607
3	3.5	0.285	33	0.59484	0.59492	0.58985	0.58993
4	5.0	0.200	35	0.59384	0.59392	0.59483	0.59490
5	5.0	0.200	35	0.59381	0.59389	0.59540	0.59548
6	1.5	0.667	30	0.59367	0.59375	0.59383	0.59390
7	5.0	0.200	35	0.59378	0.59386	0.59285	0.59293
8	2.0	0.500	33	0.59371	0.59379	0.59520	0.59527
9	2.5	0.400	33	0.59317	0.59317	0.59375	0.59382
10	5.0	0.200	35	0.59390	0.59398	0.59348	0.59355

Nilai AUC data *testing* paling tinggi pada Tabel 4.8 adalah 0,59640 pada data ke-1. Sementara pada data *training*, nilai AUC tertinggi ada pada data ke-3. Namun pemilihan tetep didasarkan pada hasil AUC data *testing*. Nilai parameter λ yang optimum pada model ini adalah 0,5 dengan 33 variabel yang terpilih. Variabel yang terpilih pada masing-masing skenario data berkisar antara 30 sampai 35 variabel dari total 36 variabel. Hal ini menandakan berdasarkan metode *lasso* regresi logistik, jumlah variabel yang tidak signifikan sedikit. Sementara itu penggunaan parameter *c* yang kecil membuat variabel yang terpilih sedikit dimana sedikitnya variabel yang terpilih membuat nilai AUC semakin kecil.

4.3.3 *Elastic-Net* Regresi Logistik

Penelitian ini menggunakan nilai parameter yang dicobakan adalah *C*=[0.1, 0.5, 1.5, 2, 2.5] dan γ=[0.6, 0.7, 0.8, 0.9]. Nilai *C* merupakan nilai yang berbanding terbalik dengan *regularized parameter* yang mengatur banyak sedikitnya variabel yang terpilih.

Kedua parameter tersebut dikombinasi untuk mendapatkan nilai AUC pada data *testing* paling tinggi sehingga didapatkan parameter yang paling optimum.

Tabel 4.9 Nilai Kebaikan Model *Elastic-Net* Regresi Logistik

Data	C	λ	γ	p*	Training		Testing	
					Akurasi	AUC	Akurasi	AUC
1	2.5	0.4	0.9	33	0.5910	0.5911	0.5926	0.5925
2	2.5	0.4	0.9	33	0.5911	0.5915	0.5937	0.5925
3	2.5	0.4	0.9	31	0.5923	0.5911	0.5868	0.5923
4	2.5	0.4	0.9	33	0.5917	0.5915	0.5914	0.5926
5	2.5	0.4	0.9	33	0.5911	0.5910	0.5915	0.5924
6	2.5	0.4	0.9	32	0.5915	0.5914	0.5914	0.5926
7	2.5	0.4	0.9	34	0.5915	0.5912	0.5911	0.5927
8	2.5	0.4	0.9	34	0.5915	0.5913	0.5911	0.5921
9	2.5	0.4	0.9	34	0.5908	0.5909	0.5920	0.5920
10	2.5	0.4	0.9	34	0.5913	0.5914	0.5917	0.5918

Model terbaik berdasarkan Tabel 4.9 adalah model pada skenario ke-7 dengan nilai AUC data testing tertinggi yaitu sebesar 0,5927. Variabel yang terpilih pada model ini adalah sebanyak 34 dengan nilai parameter λ optimum sebesar 0,4 dan nilai γ optimum sebesar 0,9. Rata-rata nilai AUC data testing pada metode *elastic-net* regresi logistik adalah sebesar 0,5924. Nilai AUC pada data *testing* lebih tinggi daripada pada data *training* namun selisihnya sangat kecil.

4.4 Prediktor Relevan

Prediktor relevan didapatkan dari model terbaik dari masing-masing metode *regularized* yaitu *lasso* SVM, *elastic-net* SVM, *lasso* regresi logistik dan *elastic-net* regresi logistik. Prediktor yang terpilih didasarkan pada hasil estimasi koefisien. Apabila estimasi koefisien variabel bernilai 0 maka artinya variabel tersebut dinilai tidak signifikan terhadap model dan dikeluarkan dari model sehingga didapatkan variabel hasil *feature selection.*

Tabel 4.10 Estimasi Parameter Model *Regularized* SVM

Metode	Par	Koef	Metode	Par	Koef
Lasso SVM	*b*	0.589	*Elastic-Net* SVM	*b*	0.233
	w_1	-0.035		w_1	-0.252
	w_2	-0.008		w_2	-0.044
	w_3	-0.546		w_3	-2.701
	w_4	-1.475		w_4	-1.283
	w_5	0.000*		w_5	0.000*
	w_6	3.046		w_6	0.000*
	w_7	1.317		w_7	-1.867
	w_8	2.436		w_8	-1.773
	w_9	-6.570		w_9	-2.479
	w_{10}	-0.004		w_{10}	0.122
	w_{11}	-1.046		w_{11}	0.069
	w_{12}	1.413		w_{12}	0.101
	w_{13}	0.442		w_{13}	0.781
	w_{14}	-0.736		w_{14}	-0.252
	w_{15}	0.749		w_{15}	0.673
	w_{16}	-0.072		w_{16}	0.000*
	w_{17}	-0.178		w_{17}	0.000*
	w_{18}	0.170		w_{18}	0.096
	w_{19}	-0.692		w_{19}	0.917
	w_{20}	-4.484		w_{20}	-2.005
	w_{21}	6.356		w_{21}	1.706
	w_{22}	-0.269		w_{22}	-1.299
	w_{23}	-0.015		w_{23}	0.389
	w_{24}	-0.110		w_{24}	0.108
	w_{25}	0.018		w_{25}	-0.037
	w_{26}	-0.140		w_{26}	0.126
	w_{27}	-0.120		w_{27}	-0.402
	w_{28}	-4.273		w_{28}	-3.023

Tabel 4.10 Estimasi Parameter Model *Regularized* SVM (Lanjutan)

Metode	Par	Koef	Metode	Par	Koef
Lasso SVM	w_{29}	-51.624	*Elastic-Net* SVM	w_{29}	-2.018
	w_{30}	29.204		w_{30}	1.826
	w_{31}	-1.876		w_{31}	-2.755
	w_{32}	0.461		w_{32}	2.229
	w_{33}	0.351		w_{33}	3.803
	w_{34}	-0.114		w_{34}	1.428
	w_{35}	0.092		w_{35}	1.800
	w_{36}	-0.889		w_{36}	-0.512

Tabel 4.10 menunjukkan adanya irisan yang sama antara variabel terpilih pada metode *lasso* SVM dan *elastic-net* SVM. Pada metode *lasso* SVM terdapat 35 variabel terpilih dimana variabel yang tidak signifikan adalah nilai maksimum transaksi telepon. Sementara pada *elastic-net* SVM terdapat 32 variabel terpilih dengan variabel nilai maksimum transaksi telepon, nilai total transaksi telepon, waktu minimum pembenahan kejadian *trouble* dan waktu maksimum pembenahan kejadian *trouble*.

Tabel 4.11 Estimasi Parameter Model *Regularized* Regresi Logistik

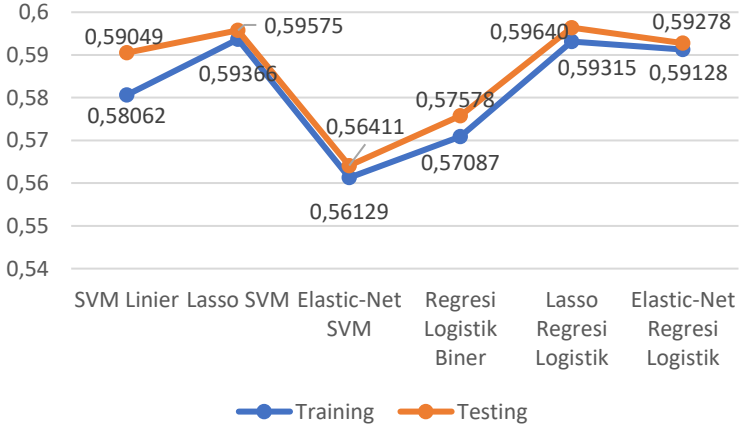
Metode	Par	Koef	Metode	Par	Koef
Lasso Regresi Logistik	*b*	1.231	*Elastic-Net* Regresi Logistik	*b*	1.051
	X_1	-0.057		X_1	-0.060
	X_2	-0.018		X_2	-0.018
	X_3	-1.168		X_3	-1.154
	X_4	-1.821		X_4	-1.471
	X_5	0.000*		X_5	1.616
	X_6	0.000*		X_6	0.000*
	X_7	1.058		X_7	1.908
	X_8	4.953		X_8	6.066
	X_9	-12.434		X_9	-14.207
	X_{10}	-0.370		X_{10}	0.054
	X_{11}	0.000*		X_{11}	0.000*

Tabel 4.11 Estimasi Parameter Model *Regularized* Regresi Logistik (Lanjutan)

Metode	Par	Koef	Metode	Par	Koef
	X_{12}	1.352		X_{12}	1.360
	X_{13}	0.891		X_{13}	0.773
	X_{14}	-1.464		X_{14}	-1.344
	X_{15}	1.618		X_{15}	1.549
	X_{16}	-0.524		X_{16}	0.086
	X_{17}	-0.447		X_{17}	-0.483
	X_{18}	0.617		X_{18}	0.355
	X_{19}	-0.173		X_{19}	-2.599
	X_{20}	-9.831		X_{20}	-15.271
	X_{21}	13.159		X_{21}	18.055
	X_{22}	-0.613		X_{22}	-0.687
	X_{23}	-0.116		X_{23}	0.065
Lasso Regresi Logistik	X_{24}	-0.258	*Elastic-Net* Regresi Logistik	X_{24}	-0.291
	X_{25}	0.099		X_{25}	0.098
	X_{26}	-0.262		X_{26}	-0.258
	X_{27}	-0.290		X_{27}	-0.214
	X_{28}	-9.384		X_{28}	-7.859
	X_{29}	-101.481		X_{29}	-79.034
	X_{30}	60.546		X_{30}	48.088
	X_{31}	-3.930		X_{31}	-3.691
	X_{32}	1.025		X_{32}	1.095
	X_{33}	1.015		X_{33}	1.188
	X_{34}	-0.415		X_{34}	-0.205
	X_{35}	0.158		X_{35}	0.336
	X_{36}	-1.863		X_{36}	-1.718

Tabel 4.11 menunjukkan hasil prediktor yang relevan terhadap model. Hasil estimasi parameter yang menunjukkan nilai 0 menunjukkan bahwa variabel tersebut tidak terpilih pada model. Estimasi model *regularized* untuk mendapatkan prediktor yang relevan didasarkan pada model yang memiliki nilai AUC paling

tinggi. Prediktor yang terpilih pada model *lasso* lebih sedikit daripada model *elastic-net* regresi logistik. Model *lasso* regresi logistik menghasilkan 33 variabel terpilih dimana variabel x_5 (nilai maksimum transaksi telepon), x_6 (total transaksi telepon) dan x_{11} (nilai maksimum penggunaan internet) tidak terpilih pada model. Sementara itu pada model *elastic-net* regresi logistik jumlah variabel yang terpilih adalah sebanyak 34 variabel dengan variabel x_6 (total transaksi telepon) dan x_{11} (nilai maksimum penggunaan internet) tidak terpilih pada model. Kedua model regresi logistik ini memiliki irisan variabel tidak terpilih yang sama. Pada model terbaik *regularized* regresi logistik yaitu *lasso*, pelanggan perempuan cenderung akan 1,06 kali telat bayar daripada pelanggan laki-laki dan apabila pelanggan menambah masa penggunaan jasa selama 6 bulan maka akan cenderung tidak akan telat membayar sebesar 1,43 kali daripada pelangggan yang tidak menambah masa penggunaan jasa.



Gambar 4.3 Perbandingan AUC

Gambar 4.3 menunjukkan bahwa nilai AUC secara keseluruhan pada data *testing* lebih tinggi daripada data *training* namun selisihnya hanya sedikit. Model terbaik yang dapat digunakan untuk memprediksi telat bayar pelanggan adalah metode *lasso* regresi logistik dengan nilai AUC sebesar 0,59640.

Prediktor relevan yang digunakan untuk memprediksi telat bayar Perusahan Telco adalah dengan tidak menyertakan variabel nilai maksimum transaksi telepon, total transaksi telepon dan nilai maksimum penggunaan internet, sehingga ada total 33 variabel relevan yang terpilih.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut :

1. Variabel prediktor yang digunakan untuk memprediksi telat bayar pelanggan Perusahaan Telco tidak dapat membedakan antara pelanggan yang telat membayar ataupun pelanggan yang tidak telat membayar dimana baik pada histogram dan boxplot mempunyai nilai yang hampir sama. Namun dari hasil uji beda *mean* menunjukkan terdapat delapan variabel kontiyu yang memiliki nilai rata-rata yang sama.
2. Metode *lasso* SVM dapat memprediksi telat bayar pelanggan dengan nilai AUC data *testing* lebih baik daripada SVM linier yaitu sebesar 0,59575 dengan 35 variabel terpilih. Sementara metode *elastic-net* SVM dapat memprediksi telat bayar pelanggan denga nilai AUC data *testing* sebesar 0,56411 dengan 32 variabel prediktor yang terpilih. Hasil kedua metode tidak menunjukkan adanya *overfitting*.
3. Hasil AUC data *testing* dalam memprediksi telat bayar pelanggan Perusahaan Telco dengan metode *lasso* regresi logistik adalah sebesar 0,59640 dan metode *elastic-net* regresi logistik sebesar 0,59278. Metode *regularized* dapat menghasilkan nilai AUC yang lebih baik daripada metode regresi logistik biner. Variabel prediktor yang terpilih pada masing-masing metode adala sebanyak 33 dan 34 variabel,. Hasil kedua metode ini juga tidak menunjukkan adanya *overfitting*.
4. Variabel prediktor relevan yang direkomendasikan untuk dimasukkan dalam pemodelan prediksi telat bayar pelanggan Perusahaan Telco ada sebanyak 33 variabel dari total 36 variabel penelitian dimana nilai maksimum transaksi telepon, total transaksi telepon dan nilai maksimum penggunaan internet tidak diikutkan dalam model. Metode *lasso* regresi logistik merupakan metode yang paling baik digunakan

dimana menghasilkan nilai AUC paling besar diantara ketiga metode lainnya.

5.2 Saran

Saran untuk penelitian selanjutnya yaitu agar dapat menggunakan metode nonlinier seperti *random forest* yang dapat memungkinkan mendapatkan hasil yang lebih baik. Penggunaan nilai parameter regularisasi perlu diperhatikan agar mendapatkan hasil yang optimum. Analisis data dengan data pengamatan yang cukup banyak dianjurkan menggunakan software *python* dengan *google colab* dibandingkan menggunakan software R.

DAFTAR PUSTAKA

Agresti, A. (2002). *Categorical Data Analysis* (2nd ed.). USA: Wiley.

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankcrptcy. *The Journal of Finance, 23*(4), 589-609.

Beaver, W. (1966). Finantial Ratios as Predictors of Failures. *Journal of Accounting Research, 4*, 71-111.

Becker, N., Toedt, G., Lichter, P., & Benner, A. (2011). Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. *BMC bioinformatics, 12*(1), 138.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature Selection via Concave Minimization and Support Vector Machines. *The Fifteenth International Conference on Machine Learning (ICML).*

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees.* Boca Raton: Chapman & Hall.

Chen, S., Härdle, W. K., & Moro, R. A. (2006). *Estimation of Default Probabilities with Support Vector Machines.* Berlin: SFB 649 Economic Risk.

Danenas, P., Garsva, G., & Gudas, S. (2011). Credit Risk E valuation Model Development Using Support Vector Based Classifier. *Procedia Computer Science*, 1699–1707.

Dubey, R., Zhou, J., Wang, Y., Thompson, P. M., & Ye, J. (2014). Analysis of Sampling Techniques for Imbalanced Data: An n=648 ADNI Study. *Neuro Image*, 220-241.

Fan, J., & Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of The American Statistical Association*, 1348-1360.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Path for Generalized Linear Models via Coordonate Descent. *Journal of Statistical Software*, 1-22.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research, 3*, 1157-1182.

Härdle, W. K., & Prastyo, D. D. (2014). Embedded Predictor Selection for Default Risk Calculation: A Southeast Asian Industry Study. In D. L. Chuen, & G. N. Gregoriou (Eds.), *Financial Markets and Sovereign Wealth Funds* (pp. 131-148). United States of America: Academic Press.

Härdle, W. K., Prastyo, D. D., & Hafner, C. M. (2014). Support Vector Machines with Evolutionary Model Selection for Default Prediction. In J. S. Racine, L. Su, & A. Ullah (Eds.), *The Oxford Handbook of Applied Nonparametric and Semiparametric Econometrics and Statistics* (pp. 346-373). USA: Oxford University Press.

Härdle, W., Moro, R. A., & Schäfer, D. (2005). *Predicting Bankruptcy with Support Vector Machines.* Berlin: SFB 649 Economic Risk.

Hastie, T., Rosset, S., Tibshirani, R., & Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 1391-1415.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression, 2nd ed.* New York: John Wiley and Sons.

Huang, J., & Ling, C. X. (2005). Using AUC and Accuracy in Evaluating Learning Algotithms. *IEEE Transactions on Knowledge and Data Engineering*, 299-310.

Laitinen, E. K. (2006). Partial Least Squares Regression in Payment Default Prediction. *Investment Management and Financial Innovations, 3*(1), 66-77.

Merwin, C. L. (1942). Financing Small Corporations in Five Manufacturing Industries. *National Bureau of Economic Research*, 1926-36.

Natasha, A. (2019). *Analisis Prediksi Risiko Kegagalan Bayar Kewajiban Menggunakan Metode Deep Support Vector Learning* . Surabaya: Departemen Statistika, Fakultas Matematika Komputasi dan Science Data Instiitut Teknologi Sepuluh Nopember.

Natasha, A., Prastyo D. D., Suhartono (2019). Credit Scoring to Classify Consumer Loan Using Machine Learning. In *AIP Conference Proceedings* (Vol. 2194, No. 1, p. 020070). AIP Publishing LLC.

Ohlson, J. A. (1980). Financial Ratios and The Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, 109-131.

Prastyo, D. D. (2015). Reguralized Support Vector Machines with Aplication to Corporate Default Prediction. *Manuscript.* Institut Teknologi Sepuluh Nopember Surabaya.

Reimeinda, V., Murni, S., & Saerang, I. (2016). Analisis Pengaruh Modal Kerja Terhadap Profitabilitas pada Industri Telekomunikasi di Indonesia. Manado: Jurnal Berkala Ilmiah Efisiensi.

Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* Cambridge: MIT Press.

Su, L., & Zhang, Y. (2014). Variable Selection in Nonparametric amd Semiparametric Regression. In J. S. Racine, L. Su, & U. Aman, *The Oxford Handbook of Applied Nonparametric and Semiparametric Ecnometrics and Statistics* (pp. 249-307). United States of America: Oxford University Press.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 267-288.

Walpole, R. (2007). *Pengantar Statistika Edisi ke-3 Terjemahan Bambang Sumantri.* Jakarta: Gramedia.

Wilson, N. (2008). *An Investigation into Payment Trends and Behavior in the UK: 1997-2007.* Leeds: CMRC.

Wu, Q., & Wang, W. (2013). Piecewise-Smooth Support Vector Machine for Classification. *Mathematical Problems in Enggineering*, 7.

Ye, K. M., & Rahman, H. A. (2010). Risk of Late Payment in the Malaysian Construction Industry. *International Journal of Mechanical and Industrial Engineering, 4*(5), 503-511.

Zhu, J., Rosset, S., Hastie, T., & Tibshirani, R. (2004). 1-norm Support Vector Machines. *Advances in Neural Information Processing System 16.*

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via The Elastic Net. *Journal of The Royal Statistics Society*, 301-320.

LAMPIRAN

Lampiran 1. Data Penelitian

Tenure	Gender	Social Economy Category	Voice Call Duration	...	Internet Payment Ratio M6	Y
8.9026	M	Urban Middle-Class	5	...	1.2	0
17.5048	M	Rural Middle-Class	6	...	0	1
36.5836	M	Rural Middle-Class	168	...	1.0009	1
36.2324	M	Urban Middle-Class	1	...	1.7018	1
8.555	M	Rural Middle-Class	1	...	0.9886	1
86.3823	M	Urban Middle-Class	311	...	0.9983	1
30.555	M	Urban Middle-Class	6	...	0	0
75.3823	F	Rural Upscale	8	...	1.0407	1
81.2939	M	Urban Middle-Class	2	...	0.9986	0
14.5215	F	Urban Middle-Class	1	...	1.6305	1
39.04	M	Urban Middle-Class	7	...	0	1
38.7473	M	Urban Middle-Class	1	...	0.9122	0
13.3501	M	Urban Middle-Class	1	...	1.097	1
...	...	...	...	...	...	...
...	...	...	...	...	...	...
3.7449	F	Rural Middle-Class	0	...	0	1
20.3638	M	Urban Middle-Class	0	...	1.7321	0
19.0753	F	Urban Middle-Class	0	...	1.0146	0
31.2467	F	Rural Middle-Class	0	...	1.1472	0

Lampiran 2. *Syntax* Eksplorasi Data

```
#Membuka directory google drive
from google.colab import drive
drive.mount('/content/gdrive')

#import package python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#Membaca Data Lengkap
data = pd.read_csv("/content/gdrive/My Drive/Data TA/Data TA Lengkap.csv")

#Membuat boxplot data kategorik
sns.boxplot(x=data['Y_RISK_NEW'],y=data['TENURE'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_CALL_TOTAL_MIN6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_CALL_TOTAL_MAX6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_CALL_TOTAL_SUM6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_DUREE_TOTAL_MIN6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_DUREE_TOTAL_MAX6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_DUREE_TOTAL_SUM6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_USAGE_MIN6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_USAGE_MAX6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_USAGE_SUM6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['ALL_TROUBLE_MIN6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['ALL_TROUBLE_MAX6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['ALL_TROUBLE_SUM6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['ALL_MTTR_MIN6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['ALL_MTTR_MAX6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['ALL_MTTR_SUM6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_MIN6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_MAX6'])
plt.show()
```

```
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_SUM6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_RATIO_M1'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_RATIO_M2'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_RATIO_M3'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_RATIO_M4'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_RATIO_M5'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['VOICE_PAYMENT_RATIO_M6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_MIN6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_MAX6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_SUM6'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_RATIO_M1'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_RATIO_M2'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_RATIO_M3'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_RATIO_M4'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_RATIO_M5'])
plt.show()
sns.boxplot(x=data['Y_RISK_NEW'],y=data['INT_PAYMENT_RATIO_M6'])
plt.show()
```

Lampiran 3. *Syntax* Metode SVM linier

```
#import package
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score, roc_auc_score, confusion_matrix

#Membaca Data
datatrain1 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct1train.csv")
datatrain2 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct2train.csv")
datatrain3 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct3train.csv")
datatrain4 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct4train.csv")
datatrain5 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct5train.csv")
datatrain6 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct6train.csv")
datatrain7 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct7train.csv")
datatrain8 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct8train.csv")
datatrain9 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct9train.csv")
datatrain10 = pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct10train.csv")

datatrain1['GENDER']=datatrain1['GENDER'].astype('category')
```

```
datatrain1['SOCIAL_ECO_CAT']=datatrain1['SOCIAL_ECO_CAT'].astype('category'
)
datatrain1['Y_RISK']=datatrain1['Y_RISK'].astype('category')
datatrain2['GENDER']=datatrain2['GENDER'].astype('category')
datatrain2['SOCIAL_ECO_CAT']=datatrain2['SOCIAL_ECO_CAT'].astype('category'
)
datatrain2['Y_RISK']=datatrain2['Y_RISK'].astype('category')
datatrain3['GENDER']=datatrain3['GENDER'].astype('category')
datatrain3['SOCIAL_ECO_CAT']=datatrain3['SOCIAL_ECO_CAT'].astype('category'
)
datatrain3['Y_RISK']=datatrain3['Y_RISK'].astype('category')
datatrain4['GENDER']=datatrain4['GENDER'].astype('category')
datatrain4['SOCIAL_ECO_CAT']=datatrain4['SOCIAL_ECO_CAT'].astype('category'
)
datatrain4['Y_RISK']=datatrain4['Y_RISK'].astype('category')
datatrain5['GENDER']=datatrain5['GENDER'].astype('category')
datatrain5['SOCIAL_ECO_CAT']=datatrain5['SOCIAL_ECO_CAT'].astype('category'
)
datatrain5['Y_RISK']=datatrain5['Y_RISK'].astype('category')
datatrain6['GENDER']=datatrain6['GENDER'].astype('category')
datatrain6['SOCIAL_ECO_CAT']=datatrain6['SOCIAL_ECO_CAT'].astype('category'
)
datatrain6['Y_RISK']=datatrain6['Y_RISK'].astype('category')
datatrain7['GENDER']=datatrain7['GENDER'].astype('category')
datatrain7['SOCIAL_ECO_CAT']=datatrain7['SOCIAL_ECO_CAT'].astype('category'
)
datatrain7['Y_RISK']=datatrain7['Y_RISK'].astype('category')
datatrain8['GENDER']=datatrain8['GENDER'].astype('category')
datatrain8['SOCIAL_ECO_CAT']=datatrain8['SOCIAL_ECO_CAT'].astype('category'
)
datatrain8['Y_RISK']=datatrain8['Y_RISK'].astype('category')
datatrain9['GENDER']=datatrain9['GENDER'].astype('category')
datatrain9['SOCIAL_ECO_CAT']=datatrain9['SOCIAL_ECO_CAT'].astype('category'
)
datatrain9['Y_RISK']=datatrain9['Y_RISK'].astype('category')
datatrain10['GENDER']=datatrain10['GENDER'].astype('category')
datatrain10['SOCIAL_ECO_CAT']=datatrain10['SOCIAL_ECO_CAT'].astype('catego
ry')
datatrain10['Y_RISK']=datatrain10['Y_RISK'].astype('category')

xtrain1 = datatrain1.drop(['Y_RISK'], axis=1)
xtrain2 = datatrain2.drop(['Y_RISK'], axis=1)
xtrain3 = datatrain3.drop(['Y_RISK'], axis=1)
xtrain4 = datatrain4.drop(['Y_RISK'], axis=1)
xtrain5 = datatrain5.drop(['Y_RISK'], axis=1)
xtrain6 = datatrain6.drop(['Y_RISK'], axis=1)
xtrain7 = datatrain7.drop(['Y_RISK'], axis=1)
xtrain8 = datatrain8.drop(['Y_RISK'], axis=1)
xtrain9 = datatrain9.drop(['Y_RISK'], axis=1)
xtrain10 = datatrain10.drop(['Y_RISK'], axis=1)
```

```
ytrain1 = datatrain1['Y_RISK']
ytrain2 = datatrain2['Y_RISK']
ytrain3 = datatrain3['Y_RISK']
ytrain4 = datatrain4['Y_RISK']
ytrain5 = datatrain5['Y_RISK']
ytrain6 = datatrain6['Y_RISK']
ytrain7 = datatrain7['Y_RISK']
ytrain8 = datatrain8['Y_RISK']
ytrain9 = datatrain9['Y_RISK']
ytrain10 = datatrain10['Y_RISK']

datatest1=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct1test.csv")
datatest2=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct2test.csv")
datatest3=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct3test.csv")
datatest4=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct4test.csv")
datatest5=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct5test.csv")
datatest6=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct6test.csv")
datatest7=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct7test.csv")
datatest8=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct8test.csv")
datatest9=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct9test.csv")
datatest10=pd.read_csv("/content/gdrive/My Drive/Data TA/STD/ct10test.csv")

datatest1['GENDER']=datatest1['GENDER'].astype('category')
datatest1['SOCIAL_ECO_CAT']=datatest1['SOCIAL_ECO_CAT'].astype('category')
datatest1['Y_RISK']=datatest1['Y_RISK'].astype('category')
datatest2['GENDER']=datatest2['GENDER'].astype('category')
datatest2['SOCIAL_ECO_CAT']=datatest2['SOCIAL_ECO_CAT'].astype('category')
datatest2['Y_RISK']=datatest2['Y_RISK'].astype('category')
datatest3['GENDER']=datatest3['GENDER'].astype('category')
datatest3['SOCIAL_ECO_CAT']=datatest3['SOCIAL_ECO_CAT'].astype('category')
datatest3['Y_RISK']=datatest3['Y_RISK'].astype('category')
datatest4['GENDER']=datatest4['GENDER'].astype('category')
datatest4['SOCIAL_ECO_CAT']=datatest4['SOCIAL_ECO_CAT'].astype('category')
datatest4['Y_RISK']=datatest4['Y_RISK'].astype('category')
datatest5['GENDER']=datatest5['GENDER'].astype('category')
datatest5['SOCIAL_ECO_CAT']=datatest5['SOCIAL_ECO_CAT'].astype('category')
datatest5['Y_RISK']=datatest5['Y_RISK'].astype('category')
datatest6['GENDER']=datatest6['GENDER'].astype('category')
datatest6['SOCIAL_ECO_CAT']=datatest6['SOCIAL_ECO_CAT'].astype('category')
datatest6['Y_RISK']=datatest6['Y_RISK'].astype('category')
datatest7['GENDER']=datatest7['GENDER'].astype('category')
datatest7['SOCIAL_ECO_CAT']=datatest7['SOCIAL_ECO_CAT'].astype('category')
datatest7['Y_RISK']=datatest7['Y_RISK'].astype('category')
datatest8['GENDER']=datatest8['GENDER'].astype('category')
datatest8['SOCIAL_ECO_CAT']=datatest8['SOCIAL_ECO_CAT'].astype('category')
datatest8['Y_RISK']=datatest8['Y_RISK'].astype('category')
datatest9['GENDER']=datatest9['GENDER'].astype('category')
datatest9['SOCIAL_ECO_CAT']=datatest9['SOCIAL_ECO_CAT'].astype('category')
datatest9['Y_RISK']=datatest9['Y_RISK'].astype('category')
datatest10['GENDER']=datatest10['GENDER'].astype('category')
```

```
datatest10['SOCIAL_ECO_CAT']=datatest10['SOCIAL_ECO_CAT'].astype('category'
)
datatest10['Y_RISK']=datatest10['Y_RISK'].astype('category')

xtest1=datatest1.drop(['Y_RISK'],axis=1)
xtest2=datatest2.drop(['Y_RISK'],axis=1)
xtest3=datatest3.drop(['Y_RISK'],axis=1)
xtest4=datatest4.drop(['Y_RISK'],axis=1)
xtest5=datatest5.drop(['Y_RISK'],axis=1)
xtest6=datatest6.drop(['Y_RISK'],axis=1)
xtest7=datatest7.drop(['Y_RISK'],axis=1)
xtest8=datatest8.drop(['Y_RISK'],axis=1)
xtest9=datatest9.drop(['Y_RISK'],axis=1)
xtest10=datatest10.drop(['Y_RISK'],axis=1)

ytest1=datatest1['Y_RISK']
ytest2=datatest2['Y_RISK']
ytest3=datatest3['Y_RISK']
ytest4=datatest4['Y_RISK']
ytest5=datatest5['Y_RISK']
ytest6=datatest6['Y_RISK']
ytest7=datatest7['Y_RISK']
ytest8=datatest8['Y_RISK']
ytest9=datatest9['Y_RISK']
ytest10=datatest10['Y_RISK']

#Klasifikasi Menggunakan SVM Linier
linSVM1 = LinearSVC(dual=False, random_state=False)
linSVM1.fit(xtrain1,ytrain1)
ypred_train1=linSVM1.predict(xtrain1)
ypred_test1=linSVM1.predict(xtest1)
print('Coefficient of each feature:', linSVM1.coef_)
print('Feature Selection:',np.sum(linSVM1.coef_!=0))
print('Training accuracy:', linSVM1.score(xtrain1, ytrain1))
print('AUC Training',roc_auc_score(ytrain1, ypred_train1))
print('CM Training',confusion_matrix(ytrain1, ypred_train1))
print('Test accuracy:', linSVM1.score(xtest1, ytest1))
print('AUC Testing',roc_auc_score(ytest1, ypred_test1))
print('CM Testing',confusion_matrix(ytest1, ypred_test1))
print('')

linSVM2 = LinearSVC(dual=False, random_state=False)
linSVM2.fit(xtrain2,ytrain2)
ypred_train2=linSVM2.predict(xtrain2)
ypred_test2=linSVM2.predict(xtest2)
print('Coefficient of each feature:', linSVM2.coef_)
print('Feature Selection:',np.sum(linSVM2.coef_!=0))
print('Training accuracy:', linSVM2.score(xtrain2, ytrain2))
print('AUC Training',roc_auc_score(ytrain2, ypred_train2))
print('CM Training',confusion_matrix(ytrain2, ypred_train2))
print('Test accuracy:', linSVM2.score(xtest2, ytest2))
```

```
print('AUC Testing',roc_auc_score(ytest2, ypred_test2))
print('CM Testing',confusion_matrix(ytest2, ypred_test2))
print('')
…
…
…
linSVM9 = LinearSVC(dual=False, random_state=False)
linSVM9.fit(xtrain9,ytrain9)
ypred_train9=linSVM9.predict(xtrain9)
ypred_test9=linSVM9.predict(xtest9)
print('Coefficient of each feature:', linSVM9.coef_)
print('Feature Selection:',np.sum(linSVM9.coef_!=0))
print('Training accuracy:', linSVM9.score(xtrain9, ytrain9))
print('AUC Training',roc_auc_score(ytrain9, ypred_train9))
print('CM Training',confusion_matrix(ytrain9, ypred_train9))
print('Test accuracy:', linSVM9.score(xtest9, ytest9))
print('AUC Testing',roc_auc_score(ytest9, ypred_test9))
print('CM Testing',confusion_matrix(ytest9, ypred_test9))
print('')

linSVM10 = LinearSVC(dual=False, random_state=False)
linSVM10.fit(xtrain10,ytrain10)
ypred_train10=linSVM10.predict(xtrain10)
ypred_test10=linSVM10.predict(xtest10)
print('Coefficient of each feature:', linSVM10.coef_)
print('Feature Selection:',np.sum(linSVM10.coef_!=0))
print('Training accuracy:', linSVM10.score(xtrain10, ytrain10))
print('AUC Training',roc_auc_score(ytrain10, ypred_train10))
print('CM Training',confusion_matrix(ytrain10, ypred_train10))
print('Test accuracy:', linSVM10.score(xtest10, ytest10))
print('AUC Testing',roc_auc_score(ytest10, ypred_test10))
print('CM Testing',confusion_matrix(ytest10, ypred_test10))
print('')
```

Lampiran 4. *Syntax Lasso* SVM

```
#Data1
Lamda = [7,6,5.5,5,4.5,4,3.5,3,2.5,2]
for num in Lamda:
   modelSVM1 = LinearSVC(penalty='l1', C=num, dual=False,max_iter=2000,
random_state=False)
   modelSVM1.fit(xtrain1,ytrain1)
   ypred_train1=modelSVM1.predict(xtrain1)
   ypred_test1=modelSVM1.predict(xtest1)
   print('C:', num)
   print('Coefficient of each feature:', modelSVM1.coef_)
   print('Feature Selection:',np.sum(modelSVM1.coef_!=0))
   print('Training accuracy:', modelSVM1.score(xtrain1, ytrain1))
   print('AUC Training',roc_auc_score(ytrain1, ypred_train1))
   print('CM Training',confusion_matrix(ytrain1, ypred_train1))
   print('Test accuracy:', modelSVM1.score(xtest1, ytest1))
   print('AUC Testing',roc_auc_score(ytest1, ypred_test1))
```

```
    print('CM Testing',confusion_matrix(ytest1, ypred_test1))
    print('')
…
…
…
#Data10
Lamda = [7,6,5.5,5,4.5,4,3.5,3,2.5,2]
for num in Lamda:
    modelSVM10 = LinearSVC(penalty='l1', C=num, dual=False,max_iter=2000,
random_state=False)
    modelSVM10.fit(xtrain10,ytrain10)
    ypred_train10=modelSVM10.predict(xtrain10)
    ypred_test10=modelSVM10.predict(xtest10)
    print('C:', num)
    print('Coefficient of each feature:', modelSVM10.coef_)
    print('Feature Selection:',np.sum(modelSVM10.coef_!=0))
    print('Training accuracy:', modelSVM10.score(xtrain10, ytrain10))
    print('AUC Training',roc_auc_score(ytrain10, ypred_train10))
    print('CM Training',confusion_matrix(ytrain10, ypred_train10))
    print('Test accuracy:', modelSVM10.score(xtest10, ytest10))
    print('AUC Testing',roc_auc_score(ytest10, ypred_test10))
    print('CM Testing',confusion_matrix(ytest10, ypred_test10))
    print('')
```

Lampiran 5. *Elastic-Net* SVM

```
#Import package
from sklearn import linear_model

#Data1
lambda1 = [ 0.0009,0.0007,0.0005,0.0003,0.0001]
lambda2 = [ 0.8,0.6,0.4,0.2]
for num1 in lambda1:
  for num2 in lambda2:
   SVM1=linear_model.SGDClassifier(loss='hinge',penalty='elasticnet',alpha=num1,l1_
ratio=num2, max_iter=500,random_state=False)
   SVM1.fit(xtrain1,ytrain1)
   ypred_train1=SVM1.predict(xtrain1)
   ypred_test1=SVM1.predict(xtest1)
   print('C:', num1)
   print('C:', num2)
   print('Coefficient of each feature:', SVM1.coef_)
   print('Feature Selection:',np.sum(SVM1.coef_!=0))
   print('Training accuracy:', SVM1.score(xtrain1, ytrain1))
   print('AUC Training',roc_auc_score(ytrain1, ypred_train1))
   print('CM Training',confusion_matrix(ytrain1, ypred_train1))
   print('Test accuracy:', SVM1.score(xtest1, ytest1))
   print('AUC Testing',roc_auc_score(ytest1, ypred_test1))
   print('CM Testing',confusion_matrix(ytest1, ypred_test1))
   print('')
…
…
```

```
…
#Data10
lambda1 = [ 0.0009,0.0007,0.0005,0.0003,0.0001]
lambda2 = [ 0.8,0.6,0.4,0.2]
for num1 in lambda1:
  for num2 in lambda2:
    SVM10=linear_model.SGDClassifier(loss='hinge',penalty='elasticnet',alpha=num1,l
1_ratio=num2, max_iter=500,random_state=False)
    SVM10.fit(xtrain10,ytrain10)
    ypred_train10=SVM10.predict(xtrain10)
    ypred_test10=SVM10.predict(xtest10)
    print('C:', num1)
    print('C:', num2)
    print('Coefficient of each feature:', SVM10.coef_)
    print('Feature Selection:',np.sum(SVM10.coef_!=0))
    print('Training accuracy:', SVM10.score(xtrain10, ytrain10))
    print('AUC Training',roc_auc_score(ytrain10, ypred_train10))
    print('CM Training',confusion_matrix(ytrain10, ypred_train10))
    print('Test accuracy:', SVM10.score(xtest10, ytest10))
    print('AUC Testing',roc_auc_score(ytest10, ypred_test10))
    print('CM Testing',confusion_matrix(ytest10, ypred_test10))
    print('')
```

Lampiran 6. *Syntax* Regresi Logistik Biner

```
#import package
from sklearn.linear_model import LogisticRegression
#Membaca data regresi logistik
ytrain1 = ytrain1.replace(-1, 0)
ytrain2 = ytrain2.replace(-1, 0)
ytrain3 = ytrain3.replace(-1, 0)
ytrain4 = ytrain4.replace(-1, 0)
ytrain5 = ytrain5.replace(-1, 0)
ytrain6 = ytrain6.replace(-1, 0)
ytrain7 = ytrain7.replace(-1, 0)
ytrain8 = ytrain8.replace(-1, 0)
ytrain9 = ytrain9.replace(-1, 0)
ytrain10 = ytrain10.replace(-1, 0)

ytest1 = ytest1.replace(-1, 0)
ytest2 = ytest2.replace(-1, 0)
ytest3 = ytest3.replace(-1, 0)
ytest4 = ytest4.replace(-1, 0)
ytest5 = ytest5.replace(-1, 0)
ytest6 = ytest6.replace(-1, 0)
ytest7 = ytest7.replace(-1, 0)
ytest8 = ytest8.replace(-1, 0)
ytest9 = ytest9.replace(-1, 0)
ytest10 = ytest10.replace(-1, 0)

ytrain1=ytrain1.astype('category')
```

```
ytrain2=ytrain2.astype('category')
ytrain3=ytrain3.astype('category')
ytrain4=ytrain4.astype('category')
ytrain5=ytrain5.astype('category')
ytrain6=ytrain6.astype('category')
ytrain7=ytrain7.astype('category')
ytrain8=ytrain8.astype('category')
ytrain9=ytrain9.astype('category')
ytrain10=ytrain10.astype('category')

ytest1=ytest1.astype('category')
ytest2=ytest2.astype('category')
ytest3=ytest3.astype('category')
ytest4=ytest4.astype('category')
ytest5=ytest5.astype('category')
ytest6=ytest6.astype('category')
ytest7=ytest7.astype('category')
ytest8=ytest8.astype('category')
ytest9=ytest9.astype('category')
ytest10=ytest10.astype('category')

dummy_ranks = pd.get_dummies(xtrain1['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x1=xtrain1[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain1.columns[2:]
x1_kontinyu=xtrain1[kontinyu]
trainx1=pd.concat([x1,x1_kontinyu],axis=1)
trainx1=trainx1.astype('float64')

dummy_ranks = pd.get_dummies(xtrain2['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x2=xtrain2[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain2.columns[2:]
x2_kontinyu=xtrain2[kontinyu]
trainx2=pd.concat([x2,x2_kontinyu],axis=1)
trainx2=trainx2.astype('float64')

dummy_ranks = pd.get_dummies(xtrain3['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x3=xtrain3[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain3.columns[2:]
x3_kontinyu=xtrain3[kontinyu]
trainx3=pd.concat([x3,x3_kontinyu],axis=1)
trainx3=trainx3.astype('float64')

dummy_ranks = pd.get_dummies(xtrain4['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
```

```
x4=xtrain4[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain4.columns[2:]
x4_kontinyu=xtrain4[kontinyu]
trainx4=pd.concat([x4,x4_kontinyu],axis=1)
trainx4=trainx4.astype('float64')

dummy_ranks = pd.get_dummies(xtrain5['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x5=xtrain5[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain5.columns[2:]
x5_kontinyu=xtrain5[kontinyu]
trainx5=pd.concat([x5,x5_kontinyu],axis=1)
trainx5=trainx5.astype('float64')

dummy_ranks = pd.get_dummies(xtrain6['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x6=xtrain6[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain6.columns[2:]
x6_kontinyu=xtrain6[kontinyu]
trainx6=pd.concat([x6,x6_kontinyu],axis=1)
trainx6=trainx6.astype('float64')

dummy_ranks = pd.get_dummies(xtrain7['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x7=xtrain7[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain7.columns[2:]
x7_kontinyu=xtrain7[kontinyu]
trainx7=pd.concat([x7,x7_kontinyu],axis=1)
trainx7=trainx7.astype('float64')

dummy_ranks = pd.get_dummies(xtrain8['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x8=xtrain8[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain8.columns[2:]
x8_kontinyu=xtrain8[kontinyu]
trainx8=pd.concat([x8,x8_kontinyu],axis=1)
trainx8=trainx8.astype('float64')

dummy_ranks = pd.get_dummies(xtrain9['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x9=xtrain9[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain9.columns[2:]
x9_kontinyu=xtrain9[kontinyu]
trainx9=pd.concat([x9,x9_kontinyu],axis=1)
trainx9=trainx9.astype('float64')
```

```
dummy_ranks = pd.get_dummies(xtrain10['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO
_CAT')
cols_to_keep = ['GENDER']
x10=xtrain10[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain10.columns[2:]
x10_kontinyu=xtrain10[kontinyu]
trainx10=pd.concat([x10,x10_kontinyu],axis=1)
trainx10=trainx10.astype('float64')

dummy_ranks = pd.get_dummies(xtest1['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
cols_to_keep = ['GENDER']
x1=xtest1[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtest1.columns[2:]
x1_kontinyu=xtest1[kontinyu]
testx1=pd.concat([x1,x1_kontinyu],axis=1)
testx1=testx1.astype('float64')
…
…
…
dummy_ranks = pd.get_dummies(xtest10['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO
_CAT')
cols_to_keep = ['GENDER']
x10=xtest10[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtest10.columns[2:]
x10_kontinyu=xtest10[kontinyu]
testx10=pd.concat([x10,x10_kontinyu],axis=1)
testx10=testx10.astype('float64')

#Klasifikasi Metode Regresi Logistik Biner
reglog1 = LogisticRegression(solver='saga',max_iter=1000, random_state=False)
reglog1.fit(trainx1,ytrain1)
ypred_train1=reglog1.predict(trainx1)
ypred_test1=reglog1.predict(testx1)
print('Training accuracy:', reglog1.score(trainx1, ytrain1))
print('AUC Training',roc_auc_score(ytrain1, ypred_train1))
print('CM Training',confusion_matrix(ytrain1, ypred_train1))
print('Test accuracy:', reglog1.score(testx1, ytest1))
print('AUC Testing',roc_auc_score(ytest1, ypred_test1))
print('CM Testing',confusion_matrix(ytest1, ypred_test1))
print('')

…
…
…
reglog10 = LogisticRegression(solver='saga',max_iter=1000,random_state=False)
reglog10.fit(trainx10,ytrain10)
ypred_train10=reglog10.predict(trainx10)
ypred_test10=reglog10.predict(testx10)
print('Training accuracy:', reglog10.score(trainx10, ytrain10))
print('AUC Training',roc_auc_score(ytrain10, ypred_train10))
```

```
print('CM Training',confusion_matrix(ytrain10, ypred_train10))
print('Test accuracy:', reglog10.score(testx10, ytest10))
print('AUC Testing',roc_auc_score(ytest10, ypred_test10))
print('CM Testing',confusion_matrix(ytest10, ypred_test10))
print('')

#Estimasi parameter model RLB Data 8
import statsmodels.api as sm
dummy_ranks = pd.get_dummies(xtrain1['SOCIAL_ECO_CAT'], prefix='SOCIAL_ECO_
CAT')
x8=sm.add_constant(xtrain8)
cols_to_keep = ['const', 'GENDER']
x8=x8[cols_to_keep].join(dummy_ranks.ix[:, 'SOCIAL_ECO_CAT_2':])
kontinyu=xtrain8.columns[2:]
x8_kontinyu=xtrain8[kontinyu]
trainlog8=pd.concat([ytrain8,x8,x8_kontinyu],axis=1)
trainlog8=trainlog8.astype('float64')
logxtrain8=trainlog8.columns[1:]
reglog8 = sm.Logit(trainlog8['Y_RISK'],trainlog8[logxtrain8])
result8=reglog8.fit()
print(result8.summary())

#Backward Selection
trainlog8=trainlog8.drop(['VOICE_PAYMENT_RATIO_M2'], axis=1)
logxtrain8=trainlog8.columns[1:]
reglog8 = sm.Logit(trainlog8['Y_RISK'],trainlog8[logxtrain8])
result8=reglog8.fit()
print(result8.summary())

trainlog8=trainlog8.drop(['ALL_MTTR_MIN6'], axis=1)
logxtrain8=trainlog8.columns[1:]
reglog8 = sm.Logit(trainlog8['Y_RISK'],trainlog8[logxtrain8])
result8=reglog8.fit()
print(result8.summary())
…
…
…
trainlog8=trainlog8.drop(['VOICE_PAYMENT_MIN6'], axis=1)
logxtrain8=trainlog8.columns[1:]
reglog8 = sm.Logit(trainlog8['Y_RISK'],trainlog8[logxtrain8])
result8=reglog8.fit()
print(result8.summary())

logxtrain8=trainlog8.columns[2:]
trainx8=trainlog8[logxtrain8]
testx8=testx8[['GENDER','SOCIAL_ECO_CAT_2','SOCIAL_ECO_CAT_3','SOCIAL_E
CO_CAT_4','SOCIAL_ECO_CAT_5','SOCIAL_ECO_CAT_6','SOCIAL_ECO_CAT_7','
TENURE','VOICE_CALL_TOTAL_MIN6','VOICE_DUREE_TOTAL_MIN6',
        'VOICE_DUREE_TOTAL_MAX6','VOICE_DUREE_TOTAL_SUM6','ALL_T
ROUBLE_MIN6','ALL_TROUBLE_MAX6','ALL_TROUBLE_SUM6','VOICE_PAYME
NT_MAX6','VOICE_PAYMENT_SUM6',
```

```
            'VOICE_PAYMENT_RATIO_M1','VOICE_PAYMENT_RATIO_M3','VOICE
_PAYMENT_RATIO_M5','VOICE_PAYMENT_RATIO_M6','INT_PAYMENT_MIN6','
INT_PAYMENT_MAX6',
            'INT_PAYMENT_SUM6','INT_PAYMENT_RATIO_M1','INT_PAYMENT_R
ATIO_M2','INT_PAYMENT_RATIO_M3','INT_PAYMENT_RATIO_M4','INT_PAYM
ENT_RATIO_M6']]
reglog8 = LogisticRegression(solver='saga',max_iter=1000,random_state=False)
reglog8.fit(trainx8,ytrain8)
ypred_train8=reglog8.predict(trainx8)
ypred_test8=reglog8.predict(testx8)
print('Training accuracy:', reglog8.score(trainx8, ytrain8))
print('AUC Training',roc_auc_score(ytrain8, ypred_train8))
print('CM Training',confusion_matrix(ytrain8, ypred_train8))
print('Test accuracy:', reglog8.score(testx8, ytest8))
print('AUC Testing',roc_auc_score(ytest8, ypred_test8))
print('CM Testing',confusion_matrix(ytest8, ypred_test8))
print('')
```

Lampiran 7. *Syntax Lasso* Regresi Logistik

```
#Data1
Lamda = [5,4.5,4,3.5,3,2.5,2,1.5,1,0.5]
for num in Lamda:
    modellogit1 = LogisticRegression(penalty='l1', C=num, solver='saga',max_iter=125
0,random_state=False)
    modellogit1.fit(xtrain1,ytrain1)
    ypred_train1=modellogit1.predict(xtrain1)
    ypred_test1=modellogit1.predict(xtest1)
    print('C:', num)
    print('Intercept',modellogit1.intercept_)
    print('Coefficient of each feature:', modellogit1.coef_)
    print('Feature Selection:',np.sum(modellogit1.coef_!=0))
    print('Training accuracy:', modellogit1.score(xtrain1, ytrain1))
    print('AUC Training',roc_auc_score(ytrain1, ypred_train1))
    print('CM Training',confusion_matrix(ytrain1, ypred_train1))
    print('Test accuracy:', modellogit1.score(xtest1, ytest1))
    print('AUC Testing',roc_auc_score(ytest1, ypred_test1))
    print('CM Testing',confusion_matrix(ytest1, ypred_test1))
    print('')

#Data2
Lamda = [5,4.5,4,3.5,3,2.5,2,1.5,1,0.5]
for num in Lamda:
    modellogit2 = LogisticRegression(penalty='l1', C=num, solver='saga',max_iter=110
0,random_state=False)
    modellogit2.fit(xtrain2,ytrain2)
    ypred_train2=modellogit2.predict(xtrain2)
    ypred_test2=modellogit2.predict(xtest2)
    print('C:', num)
    print('Coefficient of each feature:', modellogit2.coef_)
    print('Feature Selection:',np.sum(modellogit2.coef_!=0))
```

```
    print('Training accuracy:', modellogit2.score(xtrain2, ytrain2))
    print('AUC Training',roc_auc_score(ytrain2, ypred_train2))
    print('CM Training',confusion_matrix(ytrain2, ypred_train2))
    print('Test accuracy:', modellogit2.score(xtest2, ytest2))
    print('AUC Testing',roc_auc_score(ytest2, ypred_test2))
    print('CM Testing',confusion_matrix(ytest2, ypred_test2))
    print('')
…
…
…

#Data10
Lamda = [5,4.5,4,3.5,3,2.5,2,1.5,1,0.5]
for num in Lamda:
    modellogit10 = LogisticRegression(penalty='l1', C=num, solver='saga',max_iter=11
00,random_state=False)
    modellogit10.fit(xtrain10,ytrain10)
    ypred_train10=modellogit10.predict(xtrain10)
    ypred_test10=modellogit10.predict(xtest10)
    print('C:', num)
    print('Coefficient of each feature:', modellogit10.coef_)
    print('Feature Selection:',np.sum(modellogit10.coef_!=0))
    print('Training accuracy:', modellogit10.score(xtrain10, ytrain10))
    print('AUC Training',roc_auc_score(ytrain10, ypred_train10))
    print('CM Training',confusion_matrix(ytrain10, ypred_train10))
    print('Test accuracy:', modellogit10.score(xtest10, ytest10))
    print('AUC Testing',roc_auc_score(ytest10, ypred_test10))
    print('CM Testing',confusion_matrix(ytest10, ypred_test10))
    print('')
```

Lampiran 8. *Syntax Elastic-Net* Regresi Logistik

```
#Data 1
lamda = [ 2.5,2,1.5,0.5,0.1]
gama = [0.9,0.8,0.7,0.6]
for num1 in lamda:
  for num2 in gama:
    logit1 = LogisticRegression(penalty='elasticnet', C=num1, solver='saga',l1_ratio=nu
m2,max_iter=1000,random_state=False)
    logit1.fit(xtrain1, ytrain1)
    ypred_train1=logit1.predict(xtrain9)
    ypred_test1=logit1.predict(xtest9)
    print('alpha:', num1)
    print('gama:', num2)
    print('Coefficient of each feature:', logit1.coef_)
    print('Feature Selection:',np.sum(logit1.coef_!=0))
    print('Training accuracy:', logit1.score(xtrain1, ytrain1))
    print('AUC Training',roc_auc_score(ytrain1, ypred_train1))
    print('CM Training',confusion_matrix(ytrain1, ypred_train1))
    print('Test accuracy:', logit1.score(xtest1, ytest1))
    print('AUC Testing',roc_auc_score(ytest1, ypred_test1))
    print('CM Testing',confusion_matrix(ytest1, ypred_test1))
```

```
    print('')
…
…
…
#Data 10
lamda = [ 2.5,2,1.5,0.5,0.1]
gama = [0.9,0.8,0.7,0.6]
for num1 in lamda:
  for num2 in gama:
    logit10 = LogisticRegression(penalty='elasticnet', C=num1, solver='saga',l1_ratio=n
um2,max_iter=1000,random_state=False)
    logit10.fit(xtrain10, ytrain10)
    ypred_train10=logit10.predict(xtrain10)
    ypred_test10=logit10.predict(xtest10)
    print('alpha:', num1)
    print('gama:', num2)
    print('Coefficient of each feature:', logit10.coef_)
    print('Feature Selection:',np.sum(logit10.coef_!=0))
    print('Training accuracy:', logit10.score(xtrain10, ytrain10))
    print('AUC Training',roc_auc_score(ytrain10, ypred_train10))
    print('CM Training',confusion_matrix(ytrain10, ypred_train10))
    print('Test accuracy:', logit10.score(xtest10, ytest10))
    print('AUC Testing',roc_auc_score(ytest10, ypred_test10))
    print('CM Testing',confusion_matrix(ytest10, ypred_test10))
    print('')
```

Lampiran 9. *Output Lasso* SVM

```
#data1

C: 7
Coefficient of each feature: [[-2.83746588e-02 -8.57002909e-03 -5.54946220e-01 -
1.88581805e+00
   1.35227986e+01 -4.57239726e-01  1.42596078e+00  2.18702296e+00
  -6.27509773e+00 -2.65512456e-01 -7.90852185e+00  8.38947107e+00
   4.45620884e-01 -7.31404453e-01  7.25584476e-01 -2.09433601e-01
  -9.11767286e-01  1.14633346e+00 -6.63177452e-01 -5.14250627e+00
   6.92765980e+00 -2.69806047e-01 -6.73589984e-02 -1.31269064e-01
   3.89064637e-02 -1.38733010e-01 -1.34960352e-01 -4.60842069e+00
  -3.79324826e+01  2.73753070e+01 -1.58800409e+00  6.06443669e-01
   6.17700557e-01 -2.42958227e-01  1.45609337e-01 -8.23992788e-01]]
Feature Selection: 36
Training accuracy: 0.59228125
AUC Training 0.5923635620870082
CM Training [[42829 37285]
 [27950 51936]]
Test accuracy: 0.5949
AUC Testing 0.5949712361436228
CM Testing [[10897  9131]
 [ 7073 12899]]

C: 6
```

```
Coefficient of each feature: [[-2.83716905e-02 -8.56988012e-03 -5.54970229e-01 -
1.90368502e+00
  1.26578128e+01  0.00000000e+00  1.42596019e+00  2.20479295e+00
 -6.29176708e+00 -2.62861311e-01 -7.40227364e+00  7.90062127e+00
  4.45391688e-01 -7.31072091e-01  7.25964824e-01 -2.09792360e-01
 -9.03334820e-01  1.13568654e+00 -6.57887961e-01 -5.13169944e+00
  6.91563034e+00 -2.69715993e-01 -6.72809175e-02 -1.31222163e-01
  3.87382888e-02 -1.38576015e-01 -1.34991579e-01 -4.60731332e+00
 -3.79252895e+01  2.73681024e+01 -1.58819600e+00  6.06319670e-01
  6.17797371e-01 -2.42744304e-01  1.45634639e-01 -8.23894839e-01]]
Feature Selection: 35
Training accuracy: 0.59223125
AUC Training 0.5923136510481577
CM Training [[42820 37294]
 [27949 51937]]
Test accuracy: 0.594925
AUC Testing 0.5949962711926915
CM Testing [[10897  9131]
 [ 7072 12900]]

C: 5.5
Coefficient of each feature: [[-2.83683304e-02 -8.57028250e-03 -5.55000697e-01 -
1.88890768e+00
  1.24490322e+01  0.00000000e+00  1.41473694e+00  2.20393043e+00
 -6.27923769e+00 -2.61166892e-01 -7.08215716e+00  7.59161578e+00
  4.45245417e-01 -7.30893091e-01  7.26269635e-01 -2.10081429e-01
 -8.97483474e-01  1.12827258e+00 -6.53682399e-01 -5.12538657e+00
  6.90821724e+00 -2.69702105e-01 -6.72204506e-02 -1.31194377e-01
  3.86182340e-02 -1.38462663e-01 -1.35037341e-01 -4.60664430e+00
 -3.79207337e+01  2.73635828e+01 -1.58829792e+00  6.06201822e-01
  6.17833626e-01 -2.42624061e-01  1.45615309e-01 -8.23746337e-01]]
Feature Selection: 35
Training accuracy: 0.592225
AUC Training 0.592307374316662
CM Training [[42821 37293]
 [27951 51935]]
Test accuracy: 0.595
AUC Testing 0.5950713063397604
CM Testing [[10898  9130]
 [ 7070 12902]]

C: 5
Coefficient of each feature: [[-2.83636423e-02 -8.56929570e-03 -5.55052621e-01 -
1.87145463e+00
  1.22000894e+01  0.00000000e+00  1.40202374e+00  2.20293722e+00
 -6.26492077e+00 -2.59050826e-01 -6.69356389e+00  7.21659447e+00
  4.45033304e-01 -7.30650195e-01  7.26634850e-01 -2.10323863e-01
 -8.90818189e-01  1.11970413e+00 -6.50193793e-01 -5.11867076e+00
  6.90128165e+00 -2.69643809e-01 -6.71884187e-02 -1.31134421e-01
  3.84595964e-02 -1.38376260e-01 -1.35060345e-01 -4.60546117e+00
 -3.79112708e+01  2.73551918e+01 -1.58798167e+00  6.06433667e-01
```

```
  6.18070378e-01 -2.42176168e-01  1.45786563e-01 -8.23439232e-01]]
Feature Selection: 35
Training accuracy: 0.592275
AUC Training 0.5923573031680487
CM Training [[42829 37285]
 [27951 51935]]
Test accuracy: 0.595
AUC Testing 0.5950713063397604
CM Testing [[10898  9130]
 [ 7070 12902]]

dan seterusnya
```

Lampiran 10. *Output Elastic-Net* SVM

```
#data1

C: 0.0009
C: 0.8
Coefficient of each feature: [[-0.19749741 -0.04230993 -2.6324409   0.          0.          0.
  -0.00715892  0.          0.          0.          0.          0.
   0.          0.          0.          0.          0.          0.
   0.          0.          0.         -1.33132117  0.          0.51582012
   0.19605036  0.         -0.72375613 -1.30811552  0.          0.
  -1.96283118  1.10286729  3.06411595  0.8844277   1.67770677 -0.59079814]]
Feature Selection: 15
Training accuracy: 0.5514875
AUC Training 0.5516319573451434
CM Training [[36072 44042]
 [27720 52166]]
Test accuracy: 0.5534
AUC Testing 0.5535352749291389
CM Testing [[ 9151 10877]
 [ 6987 12985]]

C: 0.0009
C: 0.6
Coefficient of each feature: [[-0.21060103 -0.04791277 -2.52431452  0.          0.          0.
  -0.60611549  0.         -0.37643873  0.          0.          0.
   0.18205497  0.          0.          0.          0.          0.
   0.          0.          0.         -1.42041124  0.38433804  0.72389533
   0.25703816  0.         -0.68888937 -1.26967748  0.          0.
  -1.74664523  0.9736259   2.57577265  0.91268752  1.67514722 -0.62972101]]
Feature Selection: 18
Training accuracy: 0.554125
AUC Training 0.5542279594916503
CM Training [[38613 41501]
 [29839 50047]]
Test accuracy: 0.556075
```

```
AUC Testing 0.5561711150953855
CM Testing [[ 9764 10264]
 [ 7493 12479]]

C: 0.0009
C: 0.4
Coefficient of each feature: [[-0.20951881 -0.0491455  -2.4351145  -0.24541763  0.
0.
  -0.65490308 -0.37794366 -0.64852882  0.          0.          0.
   0.30034901  0.          0.          0.          0.          0.
   0.          0.          0.         -1.4133335   0.50095111  0.86455202
   0.35587259  0.21297196 -0.68596074 -1.23236107  0.          0.
  -1.62298417  0.95106367  2.27343902  0.87844887  1.54422708 -0.67040731]]
Feature Selection: 21
Training accuracy: 0.5545375
AUC Training 0.5546404425167486
CM Training [[38647 41467]
 [29807 50079]]
Test accuracy: 0.5563
AUC Testing 0.5563951005343971
CM Testing [[ 9783 10245]
 [ 7503 12469]]

C: 0.0009
C: 0.2
Coefficient of each feature: [[-0.21739234 -0.05441347 -2.35811801 -0.40974905  0.
0.
  -0.6932515  -0.53553498 -0.70894862  0.          0.          0.
   0.3678108   0.          0.14015852  0.          0.          0.
   0.         -0.20897644  0.         -1.38918633  0.56946283  0.94718473
   0.42449823  0.32947407 -0.67987636 -1.20582485 -0.06422788  0.
  -1.56529126  0.9407477   2.06160711  0.85890108  1.45734557 -0.70953313]]
Feature Selection: 24
Training accuracy: 0.5559625
AUC Training 0.5560385841058498
CM Training [[40269 39845]
 [31201 48685]]
Test accuracy: 0.557825
AUC Testing 0.5578936784716098
CM Testing [[10191  9837]
 [ 7850 12122]]

dan seterusnya
```

Lampiran 11. *Output* Regresi Logistik Biner

```
#Data1
Training accuracy: 0.57360625
```

AUC Training 0.5736602556701067
CM Training [[42922 37192]
 [31031 48855]]
Test accuracy: 0.571075
AUC Testing 0.5711313144173762
CM Testing [[10633 9395]
 [7762 12210]]
…
…
…
#Data10
Training accuracy: 0.573
AUC Training 0.5730554205356634
CM Training [[42794 37320]
 [31000 48886]]
Test accuracy: 0.5745
AUC Testing 0.5745501961183844
CM Testing [[10789 9239]
 [7781 12191]]

No. Observations:	160000
Df Residuals:	159958
Df Model:	41
Pseudo R-squ.:	0.03229
Log-Likelihood:	-1.0732e+05
LL-Null:	-1.1090e+05
LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.3677	0.064	21.531	0.000	1.243	1.492
GENDER	-0.0661	0.011	-6.080	0.000	-0.087	-0.045
SOCIAL_ECO_CAT_2	-0.1007	0.070	-1.433	0.152	-0.238	0.037
SOCIAL_ECO_CAT_3	-0.2760	0.056	-4.900	0.000	-0.386	-0.166
SOCIAL_ECO_CAT_4	-0.3665	0.081	-4.518	0.000	-0.525	-0.207
SOCIAL_ECO_CAT_5	-0.3814	0.180	-2.115	0.034	-0.735	-0.028
SOCIAL_ECO_CAT_6	-0.1214	0.011	-10.702	0.000	-0.144	-0.099
SOCIAL_ECO_CAT_7	-0.4106	0.300	-1.370	0.171	-0.998	0.177
TENURE	-1.1975	0.046	-26.057	0.000	-1.288	-1.107
VOICE_CALL_TOTAL_MIN6	-4.8203	1.642	-2.936	0.003	-8.038	-1.602
VOICE_CALL_TOTAL_MAX6	-9.1814	10.896	-0.843	0.399	-30.536	12.174
VOICE_CALL_TOTAL_SUM6	18.3840	13.091	1.404	0.160	-7.275	44.043
VOICE_DUREE_TOTAL_MIN6	4.4655	1.155	3.867	0.000	2.202	6.729
VOICE_DUREE_TOTAL_MAX6	6.2565	1.288	4.856	0.000	3.731	8.782
VOICE_DUREE_TOTAL_SUM6	-16.8253	2.008	-8.377	0.000	-20.762	-12.889
INT_USAGE_MIN6	-0.1927	0.336	-0.574	0.566	-0.851	0.466
INT_USAGE_MAX6	-8.4678	8.077	-1.048	0.294	-24.298	7.362
INT_USAGE_SUM6	9.8972	8.043	1.231	0.218	-5.867	25.661

ALL_TROUBLE_MIN6	0.8565	0.156	5.502	0.000	0.551	1.162
ALL_TROUBLE_MAX6	-1.2093	0.327	-3.702	0.000	-1.850	-0.569
ALL_TROUBLE_SUM6	1.1366	0.382	2.973	0.003	0.387	1.886
ALL_MTTR_MIN6	-0.1282	0.438	-0.293	0.770	-0.986	0.730
ALL_MTTR_MAX6	-1.4093	1.262	-1.117	0.264	-3.883	1.064
ALL_MTTR_SUM6	1.8665	1.599	1.167	0.243	-1.268	5.000
VOICE_PAYMENT_MIN6	-1.6422	1.122	-1.464	0.143	-3.841	0.556
VOICE_PAYMENT_MAX6	-10.1995	1.444	-7.064	0.000	-13.030	-7.369
VOICE_PAYMENT_SUM6	14.1159	1.717	8.222	0.000	10.751	17.481
VOICE_PAYMENT_RATIO_M1	-0.6477	0.101	-6.434	0.000	-0.845	-0.450
VOICE_PAYMENT_RATIO_M2	-0.0307	0.111	-0.277	0.782	-0.248	0.187
VOICE_PAYMENT_RATIO_M3	-0.2517	0.112	-2.243	0.025	-0.472	-0.032
VOICE_PAYMENT_RATIO_M4	0.0592	0.105	0.564	0.573	-0.147	0.265
VOICE_PAYMENT_RATIO_M5	-0.2331	0.093	-2.516	0.012	-0.415	-0.052
VOICE_PAYMENT_RATIO_M6	-0.2301	0.068	-3.361	0.001	-0.364	-0.096
INT_PAYMENT_MIN6	-9.7717	0.286	-34.205	0.00	-10.332	-9.212
INT_PAYMENT_MAX6	-110.1546	2.933	-37.559	0.000	-115.903	-104.406
INT_PAYMENT_SUM6	64.5317	1.636	39.436	0.000	61.324	67.739
INT_PAYMENT_RATIO_M1	-4.1003	0.171	-23.938	0.000	-4.436	-3.765
INT_PAYMENT_RATIO_M2	0.8701	0.168	5.172	0.000	0.540	1.200
INT_PAYMENT_RATIO_M3	0.7425	0.149	4.996	0.000	0.451	1.034
INT_PAYMENT_RATIO_M4	-0.4256	0.127	-3.363	0.001	-0.674	-0.178
INT_PAYMENT_RATIO_M5	0.1151	0.078	1.469	0.142	-0.038	0.269
INT_PAYMENT_RATIO_M6	-1.9622	0.104	-18.843	0.000	-2.166	-1.758

==========

#Backward Regresi Logistik
Logit Regression Results

==========

No. Observations:	160000
Df Residuals:	159970
Df Model:	29
Pseudo R-squ.:	0.03222
Log-Likelihood:	-1.0733e+05
LL-Null:	-1.1090e+05
LLR p-value:	0.000

==========

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.3979	0.060	23.450	0.000	1.281	1.515
GENDER	-0.0661	0.011	-6.079	0.000	-0.087	-0.045
SOCIAL_ECO_CAT_2	-0.1014	0.070	-1.444	0.149	-0.239	0.036
SOCIAL_ECO_CAT_3	-0.2754	0.056	-4.889	0.000	-0.386	-0.165
SOCIAL_ECO_CAT_4	-0.3666	0.081	-4.519	0.000	-0.526	-0.208
SOCIAL_ECO_CAT_5	-0.3823	0.180	-2.121	0.034	-0.736	-0.029
SOCIAL_ECO_CAT_6	-0.1213	0.011	-10.688	0.000	-0.143	-0.099
SOCIAL_ECO_CAT_7	-0.4093	0.300	-1.366	0.172	-0.997	0.178
TENURE	-1.1975	0.045	-26.439	0.000	-1.286	-1.109

VOICE_CALL_TOTAL_MIN6	-2.3865	0.634	-3.762	0.000	-3.630	-1.143
VOICE_DUREE_TOTAL_MIN6	3.0026	0.902	3.329	0.001	1.235	4.770
VOICE_DUREE_TOTAL_MAX6	5.9746	1.110	5.381	0.000	3.798	8.151
VOICE_DUREE_TOTAL_SUM6	-14.9028	1.524	-9.777	0.000	-17.891	-11.915
ALL_TROUBLE_MIN6	0.8467	0.154	5.482	0.000	0.544	1.149
ALL_TROUBLE_MAX6	-1.2965	0.319	-4.067	0.000	-1.921	-0.672
ALL_TROUBLE_SUM6	1.3488	0.337	4.007	0.000	0.689	2.009
VOICE_PAYMENT_MAX6	-9.1801	1.263	-7.270	0.000	-11.655	-6.705
VOICE_PAYMENT_SUM6	12.1624	1.037	11.725	0.000	10.129	14.195
VOICE_PAYMENT_RATIO_M1	-0.6692	0.097	-6.922	0.000	-0.859	-0.480
VOICE_PAYMENT_RATIO_M3	-0.2380	0.107	-2.218	0.027	-0.448	-0.028
VOICE_PAYMENT_RATIO_M5	-0.1336	0.064	-2.100	0.036	-0.258	-0.009
VOICE_PAYMENT_RATIO_M6	-0.2531	0.067	-3.772	0.000	-0.385	-0.122
INT_PAYMENT_MIN6	-9.9363	0.272	-36.501	0.000	-10.470	-9.403
INT_PAYMENT_MAX6	-111.9107	2.808	-39.858	0.000	-117.414	-106.408
INT_PAYMENT_SUM6	65.6694	1.534	42.800	0.000	62.662	68.677
INT_PAYMENT_RATIO_M1	-4.1434	0.165	-25.160	0.000	-4.466	-3.821
INT_PAYMENT_RATIO_M2	0.7961	0.136	5.855	0.000	0.530	1.063
INT_PAYMENT_RATIO_M3	0.7159	0.146	4.906	0.000	0.430	1.002
INT_PAYMENT_RATIO_M4	-0.3889	0.075	-5.181	0.000	-0.536	-0.242
INT_PAYMENT_RATIO_M6	-1.9546	0.104	-18.827	0.000	-2.158	-1.751

Lampiran 12. *Output Lasso* Regresi Logistik

```
#Data1
C: 5
Intercept [1.24832673]
Coefficient of each feature: [[-5.71358750e-02 -1.75322592e-02 -1.16434609e+00 -
2.35259319e+00
   3.48729089e+00  5.91680646e-01  2.06734574e+00  5.84018512e+00
  -1.42129833e+01 -4.35043159e-01  0.00000000e+00  1.68467424e+00
   9.20678376e-01 -1.48752920e+00  1.56051202e+00 -4.90903405e-01
  -1.41928170e+00  1.85214544e+00 -1.05402967e+00 -1.08790580e+01
   1.46915727e+01 -6.12401763e-01 -1.25311209e-01 -2.68060014e-01
   1.11091813e-01 -2.71575931e-01 -2.84366675e-01 -9.51916464e+00
  -1.03147285e+02  6.15788213e+01 -3.94981982e+00  1.02388528e+00
   1.00064634e+00 -4.39917101e-01  1.46733563e-01 -1.89494181e+00]]
Feature Selection: 35
Training accuracy: 0.59319375
AUC Training 0.5932743717490212
CM Training [[42997 37117]
 [27972 51914]]
Test accuracy: 0.59615
AUC Testing 0.5962210985933534
CM Testing [[10924  9104]
 [ 7050 12922]]
```

```
C: 4.5
Intercept [1.24692465]
Coefficient of each feature: [[-5.71186161e-02 -1.75343626e-02 -1.16465880e+00 -
2.21583211e+00
   2.78690041e+00  1.17298945e-01  1.95295018e+00  5.78071431e+00
  -1.40386455e+01 -4.30160722e-01  0.00000000e+00  1.65943446e+00
   9.18479871e-01 -1.48564579e+00  1.56494638e+00 -4.93282347e-01
  -1.34668213e+00  1.75969964e+00 -9.95301637e-01 -1.08016307e+01
   1.45883211e+01 -6.12493830e-01 -1.24633675e-01 -2.67360668e-01
   1.10004937e-01 -2.70764225e-01 -2.84760103e-01 -9.50872021e+00
  -1.03027229e+02  6.15015108e+01 -3.94820754e+00  1.02410291e+00
   1.00178967e+00 -4.37863264e-01  1.47547494e-01 -1.89261141e+00]]
Feature Selection: 35
Training accuracy: 0.59316875
AUC Training 0.5932493895107915
CM Training [[42994 37120]
 [27973 51913]]
Test accuracy: 0.59605
AUC Testing 0.5961210283972156
CM Testing [[10923  9105]
 [ 7053 12919]]

C: 4
Intercept [1.24525799]
Coefficient of each feature: [[-5.71008608e-02 -1.75375545e-02 -1.16502403e+00 -
2.11708997e+00
   1.84487773e+00  0.00000000e+00  1.83889739e+00  5.71334450e+00
  -1.38617339e+01 -4.24123842e-01  0.00000000e+00  1.62801669e+00
   9.15752857e-01 -1.48338075e+00  1.57045084e+00 -4.96279230e-01
  -1.25605226e+00  1.64433558e+00 -9.17526031e-01 -1.07029897e+01
   1.44511840e+01 -6.12544739e-01 -1.23772100e-01 -2.66501079e-01
   1.08736611e-01 -2.69799925e-01 -2.85272238e-01 -9.49595333e+00
  -1.02877092e+02  6.14061620e+01 -3.94637430e+00  1.02426193e+00
   1.00314862e+00 -4.35404872e-01  1.48558848e-01 -1.88972548e+00]]
Feature Selection: 34
Training accuracy: 0.59313125
AUC Training 0.5932119250597153
CM Training [[42989 37125]
 [27974 51912]]
Test accuracy: 0.59605
AUC Testing 0.5961208883969413
CM Testing [[10925  9103]
 [ 7055 12917]]

C: 3.5
Intercept [1.24314041]
Coefficient of each feature: [[-5.70792487e-02 -1.75418791e-02 -1.16548453e+00 -
2.01350276e+00
```

```
  6.16203248e-01  0.00000000e+00  1.70234855e+00  5.62991652e+00
 -1.36489232e+01 -4.16381852e-01  0.00000000e+00  1.58779503e+00
  9.12254007e-01 -1.48049367e+00  1.57749392e+00 -5.00150326e-01
 -1.13976679e+00  1.49635697e+00 -8.16404074e-01 -1.05761462e+01
  1.42728787e+01 -6.12593793e-01 -1.22660591e-01 -2.65401145e-01
  1.07137346e-01 -2.68578097e-01 -2.85937197e-01 -9.47963090e+00
 -1.02683679e+02  6.12838815e+01 -3.94407175e+00  1.02443567e+00
  1.00487760e+00 -4.32278597e-01  1.49859964e-01 -1.88602140e+00]]
Feature Selection: 34
Training accuracy: 0.593125
AUC Training 0.5932058264535814
CM Training [[42980 37134]
 [27966 51920]]
Test accuracy: 0.596225
AUC Testing 0.5962955737393245
CM Testing [[10933  9095]
 [ 7056 12916]]

dan seterusnya
```

Lampiran 13. *Output Elastic-Net* Regresi

```
#Data1
alpha: 2.5
gama: 0.9
Coefficient of each feature: [[-5.62882968e-02 -1.79623255e-02 -1.17926349e+00 -
1.89344044e+00
  0.00000000e+00  0.00000000e+00  1.00609896e+00  5.50965918e+00
 -1.27434932e+01 -2.49032856e-01  0.00000000e+00  1.49556105e+00
  8.91180534e-01 -1.45984552e+00  1.61133117e+00 -5.49966071e-01
 -7.18220883e-01  1.01296250e+00 -2.12273292e+00 -1.53003620e+01
  1.83572436e+01 -6.11543052e-01 -1.13318416e-01 -2.97212485e-01
  1.13813743e-01 -2.53454410e-01 -2.42845944e-01 -7.97134229e+00
 -7.44777370e+01  4.74386944e+01 -3.73113006e+00  1.21001176e+00
  1.34628173e+00 -3.39547586e-01  3.10492208e-01 -1.69146070e+00]]
Feature Selection: 33
Training accuracy: 0.59106875
AUC Training 0.5911763090511426
CM Training [[42720 37394]
 [28031 51855]]
Test accuracy: 0.5926
AUC Testing 0.5925819414606053
CM Testing [[10696  9332]
 [ 6968 13004]]


alpha: 2.5
gama: 0.8
```

```
Coefficient of each feature: [[-5.58400193e-02 -1.82296060e-02 -1.18742600e+00 -
1.86601943e+00
  3.07646981e-02  0.00000000e+00  5.47653283e-01  5.14447614e+00
 -1.18326804e+01 -1.69245133e-01  0.00000000e+00  1.50947753e+00
  8.86239760e-01 -1.45252793e+00  1.61850280e+00 -5.72640225e-01
 -6.92997835e-01  1.01380164e+00 -2.17572164e+00 -1.67758025e+01
  1.92051509e+01 -6.12222462e-01 -1.05343413e-01 -3.09084463e-01
  1.17811145e-01 -2.42942741e-01 -2.27552203e-01 -7.14326931e+00
 -5.96626621e+01  3.99792241e+01 -3.62293570e+00  1.30983989e+00
  1.53160653e+00 -2.86724453e-01  4.01736745e-01 -1.57270319e+00]]
Feature Selection: 34
Training accuracy: 0.58860625
AUC Training 0.5892266969672115
CM Training [[42542 37572]
 [28165 51721]]
Test accuracy: 0.592125
AUC Testing 0.5900573215123501
CM Testing [[10640  9388]
 [ 7013 12959]]

alpha: 2.5
gama: 0.7
Coefficient of each feature: [[-5.55236476e-02 -1.84252304e-02 -1.19326033e+00 -
1.85884547e+00
  3.83951018e-01  0.00000000e+00  1.62716245e-01  4.68330249e+00
 -1.09241279e+01 -1.16027863e-01  0.00000000e+00  1.51362257e+00
  8.83001699e-01 -1.44668444e+00  1.62277586e+00 -5.88941017e-01
 -6.79637400e-01  1.02071171e+00 -1.80463610e+00 -1.70036664e+01
  1.88785906e+01 -6.13494392e-01 -9.73439456e-02 -3.12672367e-01
  1.19342765e-01 -2.34198342e-01 -2.22558413e-01 -6.58115466e+00
 -5.01687426e+01  3.50426948e+01 -3.55377288e+00  1.37419094e+00
  1.65341336e+00 -2.47851527e-01  4.64809184e-01 -1.48492897e+00]]
Feature Selection: 34
Training accuracy: 0.58655
AUC Training 0.5870879635617461
CM Training [[42440 37674]
 [28405 51481]]
Test accuracy: 0.59055
AUC Testing 0.588805779059327
CM Testing [[10637  9391]
 [ 7060 12912]]

dan seterusnya
```

Lampiran 14. Surat Pernyataan Data

SURAT PERNYATAAN

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

Nama : Thalia Marda Santika

NRP : 06211640000087

menyatakan bahwa data yang digunakan dalam Tugas Akhir ini merupakan data yang diambil dari penelitian thesis S2 yaitu:

Judul : Analisis Prediksi Risiko Kegagalan Bayar Kewajiban Menggunakan Metode *Deep Support Vector Learning*

Peneliti : Azaria Natasha

Promotor : Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.

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

Menyetujui,
Pembimbing Tugas Akhir

(Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.)
NIP. 19831204 200812 1 002

Surabaya, 8 Januari 2020

(Thalia Marda Santika)
NRP. 06211640000087

BIODATA PENULIS

Thalia Marda Santika biasa dipanggil dengan nama Thalia yang merupakan anak pertama dari dua bersaudara dan dilahirkan di Kabupaten Banyuwangi pada tanggal 18 Maret 1998. Pendidikan yang telah ditempuh oleh penulis adalah SDN I Kebaman (2004-2010), SMPN 1 Srono (2010-2013), dan SMAN 1 Glagah (2013-2016). Kemudian dilanjutkan dengan menempuh pendidikan di Institut Teknologi Sepuluh Nopember Departemen Statistika. Selain dalam bidang akademik, penulis juga aktif organisasi di UKM Rara UKTK ITS sebagai Sekretaris II periode 2017/2018 dan Ikatan Himpunan Mahasiswa Statistika Indonesia (IHMSI) sebagai Kepala Bidang Pengembangan Organisasi Pusat periode 2018/2020. Selain itu, penulis juga aktif dalam mengikuti kepanitiaan yang diadakan oleh tingkat jurusan, ITS, maupun nasional seperti mentor dalam kegiatan GERIGI ITS 2018, Koor LO pada *big event* Statistika ITS yang biasa dikenal dengan Pekan Raya Statistika (PRS) 2017. Penulis juga pernah menjadi asisten dosen mata kuliah Pengantar Metode Statistika, Pengenalan Ilmu Komputer dan Program Komputer. Tidak hanya bidang akademik, penulis juga aktif di bidang seni, salah satunya pernah mewakili ITS dalam acara *Cultural Camp* di SUT University Thailand. Selama menjalani perkuliahan penulis juga berkesempatan dalam menjalani program *internship* di Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) Pusat Jakarta. Penulis juga pernah mengikuti kegiatan survei sebagai pengaplikasian ilmu statistika. Jika ingin memberikan saran, kritik, dan diskusi lebih lanjut, dapat menghubungi melalui email: thaliamarda18@gmail.com.