TUGAS AKHIR – KS184822

ANALISIS PERFORMANSI SMOTE PADA KLASIFIKASI *IMBALANCE HIGH DIMENSIONAL* DATA BERBASIS *LOGISTIC REGRESSION* (STUDI KASUS: SENYAWA OBAT KANKER)

CHARLES RUDIYANTO
NRP 062115 4000 0030

Dosen Pembimbing
Dr. rel. pol. Heri Kuswanto, S.Si., M.Si

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

(Halaman ini sengaja dikosongkan)

TUGAS AKHIR – KS184822

ANALISIS PERFORMANSI SMOTE PADA KLASIFIKASI *IMBALANCE HIGH DIMENSIONAL* DATA BERBASIS *LOGISTIC REGRESSION* (STUDI KASUS: SENYAWA OBAT KANKER)

CHARLES RUDIYANTO
NRP 062115 4000 0030

Dosen Pembimbing
Dr. rel. pol. Heri Kuswanto, S.Si., M.Si

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

(Halaman ini sengaja dikosongkan)

FINAL PROJECT – KS184822

ANALYSIS PERFORMANCE OF SMOTE IN IMBALANCE HIGH DIMENSIONAL DATA CLASSIFICATION BASED OF LOGISTIC REGRESSION (CASE STUDY: CANCER DRUG COMPOUND)

CHARLES RUDIYANTO
SN 062115 4000 0030

Supervisor
Dr. rel. pol. Heri Kuswanto, S.Si., M.Si

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

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

ANALISIS PERFORMANSI SMOTE PADA KLASIFIKASI *IMBALANCE HIGH DIMENSIONAL* DATA BERBASIS *LOGISTIC REGRESSION* (STUDI KASUS: SENYAWA OBAT KANKER)

TUGAS AKHIR

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

Oleh:
CHARLES RUDIYANTO
NRP.062115 4000 0030

Disetujui oleh Pembimbing:
Dr. rel. pol. Heri Kuswanto, S.Si., M.Si
NIP. 19820326 200312 1 004

( )

Mengetahui,
Kepala Departemen Statistika

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2019

(Halaman ini sengaja dikosongkan)

ANALISIS PERFORMANSI SMOTE PADA KLASIFIKASI *IMBALANCE HIGH DIMENSIONAL* DATA BERBASIS *LOGISTIC REGRESSION* (STUDI KASUS: SENYAWA OBAT KANKER)

Nama : Charles Rudiyanto
NRP : 062115 4000 0030
Departemen : Statistika-FMKSD-ITS
Dosen Pembimbing : Dr. rel. pol. Heri Kuswanto, S.Si., M.Si

Abstrak

Kanker merupakan salah satu penyakit penyebab kematian utama di seluruh dunia. Radioterapi adalah metode pengobatan menggunakan sinar pengion seperti sinar-X dan sinar gamma yang bertujuan untuk mematikan sel-sel kanker sebanyak mungkin dan memelihara jaringan sehat di sekitarnya. Radioterapi memiliki efek negatif, yakni dapat memperburuk kondisi pasien apabila jaringan normal disekitar sel kanker juga terkena paparan radiasi, termasuk p53 menginduksi apoptosis (kematian sel) jaringan dan sel normal. Radiasi membunuh sel-sel normal di sekitar sel-sel kanker. Dalam rangka menanggulangi efek radioterapi maka pada penelitian ini dilakukan analisis mengenai 84 komponen senyawa dengan 217 prediktor yang dapat menjadi proteksi radiasi atau radioprotector dengan melakukan dua percobaan yang dicobakan kepada sel normal serta sel yang terkena radiasi sinar gamma. Adapun metode yang akan digunakan pada penelitian ini yaitu Logistic Regression Ensemble (LORENS) dan Ensemble Logistic Regression (ELR) dengan Synthetic Minority Oversampling Techniuqe(SMOTE) dan tanpa SMOTE untuk mengklasifikasikan senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas yang dianggap baik sebagai radioprotector. Hasil analisis menunjukkan bahwa metode LORENS dengan 5 subruang threshold 0,5 menghasilkan nilai AUC terbaik sebesar 0,7.

Kata Kunci : ELR, LORENS, Proteksi Radiasi, SMOTE, Toksisitas

(Halaman ini sengaja dikosongkan)

ANALYSIS PERFORMANCE OF SMOTE IN IMBALANCE HIGH DIMENSIONAL DATA CLASSIFICATION BASED OF LOGISTIC REGRESSION (CASE STUDY: CANCER DRUG COMPOUND)

Name : Charles Rudiyanto
Student Number : 062115 4000 0030
Department : Statistics
Supervisor : Dr. rel. pol. Heri Kuswanto, S.Si., M.Si

Abstract

Cancer is one of the leading causes of death throughout the world. Radiotherapy is a method of treatment using ionizing rays such as X-rays and gamma rays that aim to kill as many cancer cells as possible and maintain healthy tissue around them. Radiotherapy has a negative effect, which can aggravate the patient's condition if the normal tissue around the cancer cell is also exposed to radiation exposure, including p53 inducing normal tissue and cell apoptosis (cell death). Radiation kills normal cells around cancer cells. In order to overcome the effects of radiotherapy, this study analyzed 84 components of compounds with 217 predictors that could be radiation protection or radioprotector by conducting two experiments that were tested on normal cells and cells affected by gamma radiation. The methods to be used in the study this is the Logistic Regression Ensemble (LORENS) and Ensemble Logistic Regression (ELR) with Synthetic Minority Oversampling Technique (SMOTE) and without SMOTE to classify cancer drug compounds to optimize radiation protection and toxicity that is considered good as a radioprotector. The results of the analysis show that the LORENS method with 5 subspace 0.5 thresholds produces the best AUC value of 0.7.

Keywords : ELR, LORENS, Radiation Protection, SMOTE, Toxicity

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji Syukur atas berkat Tuhan Yang Maha Esa yang daripada-Nya diberikan damai sejahterah sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“ Analisis Performansi Smote Pada Klasifikasi *Imbalance High Dimensional* Data Berbasis *Logistic Regression* (Studi Kasus: Senyawa Obat Kanker)”** dengan baik dan lancar.

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

1. Dr. Suhartono selaku Kepala Departemen Statistika ITS, Dr. Santi Wulan Purnami, S.Si, M.Si. selaku ketua Program Studi S1 Departemen Statistika ITS, yang telah menyediakan fasilitas dalam mengerjakan Tugas Akhir.
2. Dr. rel. pol. Heri Kuswanto, S.Si., M.Si. selaku dosen pembimbing yang telah meluangkan waktu memberikan bimbingan, saran, serta motivasi selama penyusunan Tugas Akhir berlangsung.
3. Dr. rel. pol. Dedy Dwi Prastyo, S.Si., M.Si. dan Santi Puteri Rahayu, S.Si, M.Si., Ph. D. selaku dosen penguji yang telah memberikan masukan dan bantuan dalam menyelesaikan Tugas Akhir.
4. Dr. Ir. Setiawan, M.Si selaku dosen wali yang telah memberikan motivasi, dukungan dan bimbingan selama prose belajar di Departemen Statistika.
5. Seluruh dosen dan *staff* pengajar Departemen Statistika yang telah memberikan ilmu dan motivasi dalam menjalani masa perkuliahan.
6. Orang tua dan saudara penulis yang selalu memberikan doa dan dukungan dalam menghadapi kesulitan selama penyusunan Tugas Akhir.
7. Sahabat-sahabat penulis, Dwinda Intan, Dimas Achmad, Ihsan Ananto, Habib Jazuli, Trianto Utomo, Arrafi Dwiargatra, Haidar Alvin, Triajeng Nuraisyah, dan Nesia Balqis yang senantiasa memberikan dukungan dalam menyelesaikan Tugas Akhir.

8. Teman-teman bimbingan Tugas Akhir Pak Heri yang memberikan informasi, dukungan dan semangat dalam menyelesaikan Tugas Akhir.
9. Teman-teman Sigma 26 VIVACIOUS dan teman-teman lain yang selalu memberikan semangat kepada penulis dalam penyusunan Tugas Akhir.
10. Seluruh pihak yang tidak bisa disebutkan satu-persatu 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 bermanfaat dan menambah wawasan bagi pembaca. Kritik dan saran sangat diperlukan untuk perbaikan di masa yang akan datang.

Surabaya, Juli 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

Halaman

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kanker merupakan salah satu penyakit penyebab kematian utama di seluruh dunia. Berdasarkan laporan *International Agency for Research on Cancer* (IARC) pada tahun 2018 akhir, Organisasi Kesehatan Dunia atau WHO mengestimasi terdapat 18,1 juta kasus kanker baru dan 9,6 juta kematian yang terjadi pada tahun tersebut. Pada tahun 2019 di Amerika Serikat diprediksi akan ada 1.762.450 kasus kanker baru yang didiagnosis serta lebih dari 4.800 kasus baru setiap hari. Pada tahun tersebut juga diperkirakan akan ada 606.880 kematian akibat kanker (Siegel, Miller, & Jemal, 2019). Salah satu fitur mendefinisikan kanker adalah pertumbuhan sel-sel baru secara abnormal yang tumbuh melampaui batas normal, dan yang kemudian menyerang bagian sebelah tubuh dan menyebar ke organ lain atau disebut metastasis (WHO, 2018). Kanker paru, hati, perut, kolektoral dan kanker payudara adalah penyebab kematian akibat kanker setiap tahunnya (Departemen Kesehatan RI, 2015). Kanker merupakan penyakit tidak menular yang berawal dari kerusakan materi genetika, atau DNA sel. Satu saja sel yang mengalami kerusakan genetika sudah cukup untuk menghasilkan jaringan kanker. Menurut hasil Survei Kesehatan Rumah Tangga (SKRT), proporsi penyebab kematian yang diakibatkan oleh penyakit menular menurun dari 69,49% (1980) menjadi 44,5% (2001), sedangkan untuk penyakit tidak menular (kanker) meningkat dari 25,41 (1980) menjadi 48,53% (2001). Faktor penyebab penyakit kanker meliputi perilaku seksual, infeksi, obat-obatan medis, *food additive*, merokok, radiasi, sinar UV, dan perubahan pada materi genetika. Perubahan pada genetika, atau disebut juga mutasi gen, dapat terjadi melalui berbagai mekanisme. Pertama disebabkan oleh kesalahan replikasi yang tejadi pada saat sel-sel yang aus digantikan oleh sel-sel baru. Penyebab kedua adalah mutase pada alur sel germinasi yang merupakan kesalahan genetika yang diturunkan dari gen orang tua (WCRF & AICR, 1997). Meningkatnya kasus penyakit kanker dengan tingkat kematian

yang cukup tinggi, maka perlu dilakukan pencegahan dini serta pengobatan untuk menekan tingkat mortalitas pada suatu negara. Adapun pengobatan atau penyembuhan yang dapat dilakukan pada penyakit kanker diantaranya melalui (Mangan, 2009): pembedahan (operasi), penyinaran (radioterapi) atau kemoterapi, peningkatan daya tahan tubuh dan pengobatan dengan *hormone*.

Penyinaran (radioterapi) atau kemoterapi salah satu cara dalam pengobatan kanker. Menurut Fithrony (2012), radioterapi dilakukan jika kanker telah menyebar luas dan bersifat kemosensitif atau *responsive* terhadap obat-obatan kimia, sehingga sel kanker tersebut dapat musnah. Radioterapi adalah metode pengobatan menggunakan sinar pengion seperti sinar-X dan sinar Gamma yang bertujuan untuk mematikan sel-sel kanker sebanyak mungkin dan memelihara jaringan sehat di sekitar sel kanker agar tidak mengalami kerusakan terlalu berat. Radioterapi adalah jenis terapi yang menggunakan radiasi tingkat tinggi untuk menghancurkan sel-sel kanker. Baik sel-sel normal maupun sel-sel kanker bisa dipengaruhi oleh radiasi ini. Radiasi akan merusak sel-sel kanker sehingga proses multiplikasi ataupun pembelahan sel-sel kanker akan terhambat. Tujuan radioterapi adalah untuk mengurangi dan menghilangkan rasa sakit atau tidak nyaman akibat kanker dan mengurangi risiko kekambuhan dari kanker. Menurut Morita, *et al*. (2014), radioterapi memiliki efek negatif, yakni dapat memperburuk kondisi pasien apabila jaringan normal disekitar sel kanker juga terkena paparan radiasi, termasuk p53 menginduksi apoptosis jaringan dan sel normal. p53 merupakan gen supresor tumor yang bertindak menghentikan perkembangan tumor. Singkatnya, radiasi membunuh sel-sel normal di sekitar sel-sel kanker. Hal ini dianggap bahwa p53 akan menjadi target untuk radioproteksi terapeutik dan mitigatif untuk menghindari apoptosis. Apoptosis merupakan suatu proses aktif yakni kematian sel melalui digesti enzimatik oleh dirinya sendiri dan mekanisme yang efisien untuk mengeliminasi sel yang tidak diperlukan dan mungkin berbahaya bagi tubuh sehingga dapat menyelamatkan organisme (William & Gilbert, 1991). Ariyasu, *et al*. (2014) menanggulangi efek radioterapi dengan melakukan penelitian mengenai 84 komponen senyawa yang dapat menjadi

proteksi radiasi atau radioprotector dengan melakukan 2 percobaan yang dicobakan kepada sel normal serta sel yang terkena radiasi sinar gama. Adapun percobaan pertama yaitu dengan memberikan senyawa pada sel normal untuk mengukur toksisitas, sedangkan percobaan kedua yaitu pemberian senyawa pada sel yang terkena radiasi gamma (10 Gy) dengan tujuan untuk mengukur fungsi proteksi radiasi. Tingkat kematian sel digunakan sebagai indikator pada kedua percobaan tersebut. Jika tingkat kematian sel rendah pada toksisitas dan tingkat kematian sel tinggi pada fungsi proteksi radiasi maka senyawa tersebut dapat digunakan sebagai *radioprotektor*.

Data dari penelitian Ariyatsu, *et al*. (2014), digunakan Matsumoto, *et al*. (2016) untuk memprediksi proteksi radiasi dan toksisitas dengan pendekatan *Machine Learning* menggunakan metode *random forest* (RF) dan *support vector machine* (SVM). Penelitian tersebut menunjukkan hasil SVM lebih baik dari *random forest* dalam memprediksi fungsi proteksi radiasi dengan nilai AUC 64,4%, sedangkan *random forest* lebih baik digunakan dibandingkan SVM untuk memprediksi toksisitas dengan nilai AUC 77.8%. Penelitian lain dengan menggunakan data yang sama dilakukan oleh Mubarok (2018) untuk mengklasifikasi senyawa obat kanker untuk optimasi proteksi radiasi menggunakan *naïve bayes classifier* (NBC) dan *classification and regression tree* (CART) dengan *feature selection* menggunakan *mean decrease gini* (MDG) dan didapatkan hasil CART lebih baik dibandingkan NBC dengan nilai AUC sebesar 66.3% menggunakan 10% prediktor terpenting. Sukmaputri (2018) mengklasifikasi senyawa obat kanker untuk optimasi toksisitas menggunakan *logistic regression ensembles* (LORENS) dan *ensemble of support vector machine* dengan *feature selection* menggunakan *mean decrease gini* (MDG) dan didapatkan hasil *ensemble of support vector machine* lebih akurat dibandingkan LORENS dengan nilai akurasi 78.89%.

Penelitian yang dilakukan oleh Mubarok (2018) dan Sukmaputri (2018), masing- masing untuk menentukan senyawa yang memberikan radiasi proteksi paling tinggi dan menentukan senyawa untuk mendapatkan tingkat kematian sel rendah pada

toksisitas sehingga dapat digunakan sebagai *radioprotector*. Pada penelitian ini, data yang digunakan penelitian Matsumoto, *et al*.(2016) yang selanjutnya digunakan oleh Mubarok (2018) dan Sukmaputri (2018) akan disatukan menjadi satu data set, dimana masing-masing dari variabel respon yaitu toksisitas dan proteksi radiasi akan menjadi variabel respon baru. Variabel baru dibentuk melalui pengkategorian sebagai berikut, yaitu jika pemberian senyawa pada sel normal yang bertujuan mengukur toksisitas memiliki tingkat kematian sel rendah dan pemberian senyawa pada sel yang terkena radiasi sinar gamma untuk mengukur fungsi proteksi radiasi memiliki tingkat kematian sel tinggi, maka variabel respon baru akan berkode 1, selain itu akan berkode 0. Variabel respon baru hasil pengkategorian menghasilkan kelas data yang *imbalance* atau tidak seimbang. Data yang tidak seimbang akan mempengaruhi metode klasifikasi untuk memprediksi kelas data minoritas, akurasi tinggi sering dicapai dalam klasifikasi tanpa penanganan data *imbalance* karena hanya berfokus pada data mayoritas dan untuk data minoritas dianggap sebagai data langka atau data tidak sengaja (He,*et al.*, 2009). Kelas data yang lebih sedikit biasa disebut kelas minor, sedangkan kelas data yang lebih banyak disebut kelas mayor.

Penanganan data *imbalance* dapat dilakukan dengan pendekatan level data. Pendekatan level data mencakup berbagai teknik *sampling* dan sintesis data untuk memperbaiki keadaan data yang tidak seimbang (Zhang, *et al.*, 2011). Pendekatan sampling terbagi menjadi *oversampling* dan *undersampling*. Metode dalam menangani data *imbalance* dengan pendekatan level data dengan teknik sampling salah satunya yaitu *Synthetic Minority Oversampling Technique* (SMOTE) yang diperkenalka oleh Chawla, Bowyer, Hall dan Kegelmeyer (2002). SMOTE merupakan metode *oversampling* yang digunakan untuk menambah jumlah kelas pada data minor dengan mereplikasi data secara acak agar seimbang dengan jumlah data kelas mayor. SMOTE efektif untuk menangani *overfitting* pada proses *oversampling* untuk menangani *imbalance* yang terjadi pada kelas yang lebih sedikit atau kelas minor (Chawla, *et al*, 2002). Penelitian menggunakan SMOTE telah dilakukan oleh Purnami,

et al (2016) dengan judul *Cervical Cancer Survival Prediction Using Hybrid of SMOTE, CART and Smooth Support Vector Machine,* menunjukkan bahwa mengatasi data *imbalance* menggunakan SMOTE meningkatkan performansi klasifikasi. Hasil klasifikasi dengan CART menggunakan SMOTE meningkatkan akurasi 81,00% sebelum menggunakan SMOTE dan 92.60% dengan menggunakan SMOTE, metode SSVM tanpa SMOTE menghasilkan akurasi 96.08% dan 95.91% dengan SMOTE, namun klasifikasi menggunakan TSSVM mengalami hal berbeda yaitu 95.00% tanpa SMOTE dan 87.44% menggunakan SMOTE.

High dimensional data merupakan keadaan dimana variabel prediktor jumlahnya lebih banyak dibandingkan jumlah penelitian. Penelitian ini menggunakan data yang *high dimensional*, sehingga diperlukan metode yang dapat mengatasi *high dimensional*. Salah satu metode untuk *high dimensional* yaitu *ensemble*. Metode *ensemble* mengkombinasikan sekumpulan *classifier* yang dilatih dengan tujuan untuk membuat model klasifikasi campuran yang terimprovisasi sehingga membuat *classifier ensemble* yang terbentuk lebih akurat dari pada *classifier* asalnya dalam melakukan suatu pengklasifikasian (Han, *et al*., 2012). Metode yang akan digunakan adalah *Logistic Regression Ensemble* (LORENS). LORENS dikembangkan pertama kali oleh Lim, *et al.* (2010). Lim, *et al.* (2010) mengatakan bahwa LORENS dapat meningkatkan akurasi, sensitifitas dan spesifitas dibandingkan dengan metode yang lainnya. LORENS dalam menangani *high dimensional* telah digunakan oleh Kuswanto, *et al* (2015) dengan judul *Logisti Regression Ensemble for Predicting Customer Defection with Very Large Sample Size*. Penelitian lain yaitu Klasifikasi Gen yang Terkait Sindrom Alzheimer Menggunakan Metode *Naïve Bayes Classifier, Binary Logistic Regression* dan *Logistic Regression Ensemble* yang dilakukan oleh Kuswanto dan Werdhana (2017) menunjukkan bahwa penggunaan LORENS memberikan hasil yang paling baik dibandingkan metode *Naïve Bayes* dan *Binary Logistic Regression* dengan akurasi sebesar 76,4% dan nilai AUC 77,4%.

Metode *ensemble* lain yang dapat digunakan untuk mengatasi *high dimensional* data yaitu *Ensemble Logistic Regression* (ELR). ELR diperkenalkan oleh Zakharov dan Dupont pada tahun 2011. ELR memberikan hasil klasifikasi yang stabil serta akurasi yang lebih baik dibandingkan *Random Forest* dan metode popular lainnya (Zakharov dan Dupont, 2011) Konsep ELR adalah melakukan seleksi variabel bersama-sama dengan estimasi parameter dengan tujuan untuk mengoptimalkan hasil dari klasifikasi . *Ensemble Logistic Regression* pernah digunakan untuk meneliti permasalahan *drug discovery* yang dilakukan oleh Widhianingsih (2018) dengan judul penelitian Klasifikasi Data Berdimensi Tinggi dengan Metode *Ensemble* Berbasis Regresi Logistik dalam Permasalahan *Drug Discovery*, hasil penelitian menunjukkan bahwa ELR unggul dalam mengatasi efek penambahan variabel, efek keseimbangan data dan efek multikolinearitas.

Penelitian ini menggunakan metode yang sama dengan penelitian yang telah dilakukan oleh Widhianingsih (2018) yaitu *Logistic Regression Ensemble* (LORENS) dan *Ensemble Logistic Regression* (ELR). Namun terdapat perbedaan yaitu pada data yang digunakan. Meskipun data yang digunakan bersumber dari penelitian Ariyasu , *et al*. (2014) namun data pada penelitian Widhianingsih hanya menggunakan data dengan variabel respon proteksi radiasi sedangkan pada penelitian ini menggunakan variabel respon proteksi radiasi dan toksisitas yang akan dijadikan menjadi satu variabel respon baru.

Berdasarkan kelebihan metode LORENS dan ELR dalam mengatasi *high dimensional* data, penelitian ini akan menggunakan metode tersebut dengan SMOTE sebagai metode dalam mengatasi data *imbalanced* . Hasil penelitian ini diharapkan dapat menghasilkan senyawa-senyawa yang menghasilkan fungsi radiasi proteksi tinggi dan toksisitas rendah sehingga dapat digunakan sebagai *radioprotector*.

1.2 Rumusan Masalah

Permasalahan yang terjadi dalam penemuan senyawa sebagai *radioprotector* dalam pengobatan kanker adalah menemukan senyawa yang memiliki tingkat kematian sel normal

rendah pada toksisitas dan tingkat kematian sel kanker tinggi pada fungsi proteksi radiasi. Pada penelitian ini terdapat *imbalance* dan *high dimensional* data dimana variabel prediktor memiliki jumlah yang lebih banyak dibandingkan jumlah penelitian. Dalam mengatasi hal tersebut dibutuhkan metode yang tepat, yaitu *Synthetic Minority Oversampling Technique* SMOTE) untuk mengatasi *imbalance* dan metode berbasis *ensemble* untuk *high-dimensional* data. Sehingga pada penelitian ini akan digunakan klasifikasi dengan metode *ensemble* yaitu *Logistic Regression Ensemble* (LORENS) dan *Ensemble Logistic Regression* (ELR) untuk memprediksi senyawa yang dianggap baik sebagai *radioprotector* serta membandingkan hasil klasifikasi berdasarkan kedua metode tersbut untuk menentukan metode terbaik berbasis *ensemble* dalam penemuan obat kanker.

1.3 Tujuan

Berdasarkan rumusan masalah yang telah disusun, tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mendapatkan karakteristik data senyawa obat kaker untuk optimasi proteksi radiasi dan toksisitas.
2. Memperoleh data *imbalance* menggunakan *Synthetic Minority Oversampling Technique* (SMOTE).
3. Mendapatkan ketepatan klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas dengan metode *Logistic Regression Ensembles* (LORENS) dengan SMOTE dan tanpa SMOTE.
4. Mendapatkan ketepatan klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas dengan metode *Ensemble Logistic Regression* (ELR) dengan SMOTE dan tanpa SMOTE.
5. Memperoleh perbandingan ketepatan klasifikasi terbaik dengan metode *Logistic Regression Ensembles* (LORENS) dan *Ensemble Logistic Regression* (ELR) dengan dan tanpa SMOTE.

1.4 Manfaat Penelitian

Berdasarkan tujuan yang ingin dicapai, adapun manfaat yang diharapkan dari hasil penelitian ini adalah sebagai berikut .

1. Diharapkan dari penelitian ini dapat memberikan manfaat bagi bidang kedokteran dalam menentukan senyawa yang bisa menjadi *radioprotector* dalam penyakit kanker.
2. Diharapkan penelitian ini dapat menjadi sumber referensi dalam bidang akademik dan pendidikan dalam penerapan metode *Logistic Regression Ensembles* (LORENS) dan *Ensemble Logistic Regression* (ELR).

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah *pre-processing* data hanya dilakukan pada variabel respon baru. Metode klasifikasi yang digunakan yaitu *Logistic Regression Ensembles* (LORENS) dan *Ensemble Logistic Regression* (ELR).

BAB II
TINJAUAN PUSTAKA

Bagian ini akan membahas beberapa tinjauan pustaka yang digunakan untuk menyelesaikan klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas dengan pendekatan *machine learning*.

2.1 Regresi Logistik

Regresi Logistik adalah suatu metode yang berguna mencari hubungan antara variabel respon *y* yang bersifat biner dengan variabel prediktor *x* yang bersifat polikotomus (Hosmer dan Lemeshow, 2009). Variabel respon *y* regresi logistik biner terdiri dari dua kategori yaitu kategori sukses dan kategori gagal yang masing-masing dapat disimbolkan dengan *y*=1 untuk kategori sukses dan *y*=0 untuk kategori gagal. Sehingga variabel respon *y* mengikuti distribusi Bernoulli untuk setiap observasi tunggalnya. Fungsi probabilitas untuk setiap observasinya adalah sebagai berikut:

$$f(y) = \pi^{y}(1-\pi)^{1-y}\,; y = 0,1 \qquad (2.1)$$

dengan $\pi(x)$ = P(Y=1) dan 1-$\pi(x)$ = P(Y=0). Bentuk persamaan model regresi logistiknya adalah sebagai berikut:

$$\pi(x) = \mathrm{P}(\mathrm{Y} = 1|\mathrm{X} = x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} \qquad (2.2)$$

dimana,
β_0 = konstanta
β_p = koefisien regresi
p = banyaknya variabel prediktor
dengan $\pi(x)$ memiliki rentang antara 0 sampai 1. Jika nilai $\pi(x)$ lebih dari 0,5 maka prediksi kategori positif atau berkode 1, namun apabila nilai $\pi(x)$ kurang dari 0,5 maka prediksi kategori negatif atau 0.

Model regresi logistik dapat dituliskan dalam bentuk logit. Menurut Yan dan Su (2009) terdapat bentuk alternatif dari

persamaan model regresi logistik yang merupakan transformasi logit dari $\pi(\boldsymbol{x})$. Model logitnya adalah sebagai berikut:

$$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \qquad (2.3)$$

dengan rentang nilai dari $g(x)$ adalah $-\infty \leq g(x) \leq \infty$.

Metode *Maximum Likelihood* (MLE) digunakan untuk mengestimasi parameter dalam regresi logistik. *MLE* didapatkan dengan cara memaksimumkan logaritma fungsi *likehood* (Hosmer dan Lemeshow, 2009). Parameter $\boldsymbol{\beta}$ diestimasi dengan memaksimumkan fungsi *likelihood* dan mensyaratkan bahwa data mengikuti suatu distribusi. Regresi logistik dalam setiap pengamatannya mengikuti distribusi Bernoulli, sehingga fungsi *likelihood*nya sebagai berikut

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} [\pi(\boldsymbol{x_i})]^{y_i} [1 - \pi(\boldsymbol{x_i})]^{1-y_i} \qquad (2.4)$$

MLE didapatkan dengan memaksimumkan fungsi *likelihood* pada Persamaan (2.4) dengan hasil sebagai berikut.

$$\ln L(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \ln \pi(\boldsymbol{x_i}) + (1 - y_i)(\ln(1 - \pi(\boldsymbol{x_i})))] \qquad (2.5)$$

$$\ln L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y\left(\boldsymbol{x}_i^T \boldsymbol{\beta}\right) - \ln(1 + \exp\left(\boldsymbol{x}_i^T \boldsymbol{\beta}\right))\right] \qquad (2.6)$$

Nilai $\boldsymbol{\beta}$ maksimum diperoleh dari diferensial $\ln L(\boldsymbol{\beta})$ terhadap $\boldsymbol{\beta}$ kemudian disamakan dengan nol. Persamaannya adalah sebagai berikut.

$$\begin{aligned}
\frac{\partial \ln L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \sum_{i=1}^{n} \left[y_i \boldsymbol{x}_i{}^T - \frac{\boldsymbol{x}_i \exp\left(\boldsymbol{x}_i^T \boldsymbol{\beta}\right)}{1 + \exp\left(\boldsymbol{x}_i^T \boldsymbol{\beta}\right)} \right] \\
&= \sum_{i=1}^{n} \boldsymbol{x}_i{}^T \left[y_i - \frac{\exp\left(\boldsymbol{x}_i^T \boldsymbol{\beta}\right)}{1 + \exp\left(\boldsymbol{x}_i^T \boldsymbol{\beta}\right)} \right] \\
&= \sum_{i=1}^{n} \boldsymbol{x}_i{}^T [y_i - \pi(\boldsymbol{x}_i)]
\end{aligned}$$

$$= \mathbf{X}^T(\boldsymbol{y} - \widehat{\boldsymbol{\pi}}) = 0 \tag{2.7}$$

Dikarenakan estimator dari MLE tidak *closed-form*, digunakan metode numerik untuk menyelesaikannya. Metode yang digunakan adalah *Newton Raphson* dengan persamaan sebagai berikut

$$\boldsymbol{\beta}^{(m)} = \boldsymbol{\beta}^{(m-1)} - \mathbf{H}^{-1}(\boldsymbol{\beta})\boldsymbol{g}(\boldsymbol{\beta}) \tag{2.8}$$

dimana, $\mathbf{H}^{-1}(\boldsymbol{\beta})$ merupakan matriks *Hessian* yang komponennya adalah turunan kedua dari fungsi ln-*likelihood* dan $\boldsymbol{g}(\boldsymbol{\beta})$ merupakan turunan pertama fungsi ln-*likelihood* terhadap parameter **β**. Turunan kedua dari fungsi ln-*likelihood* sebagai berikut.

$$\frac{\partial^2 \ln L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = \frac{\partial}{\partial \boldsymbol{\beta}^T}\left(\frac{\partial \ln L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right)$$

$$= \frac{\partial}{\partial \boldsymbol{\beta}^T}\left(\sum_{i=1}^{n}\left[y_i \boldsymbol{x}_i{}^T - \frac{\boldsymbol{x}_i \exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}\right]\right)$$

$$= -\sum_{i=1}^{n}\frac{\boldsymbol{x}_i\boldsymbol{x}_i{}^T\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})\left(1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})\right) - \boldsymbol{x}_i\boldsymbol{x}_i{}^T\left(\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})\right)^2}{\left(1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})\right)^2}$$

$$= -\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i{}^T\left[\frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})} - \left(\frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}\right)^2\right]$$

$$= -\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i{}^T\left(\frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}\right)\left(1 - \frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}\right)$$

$$= -\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i{}^T\,\pi(x_i)\big(1-\pi(x_i)\big)$$

$$= \mathbf{X}^T\mathbf{W}\mathbf{X} \tag{2.9}$$

dengan W adalah sebagai berikut

$$\boldsymbol{W} = \begin{bmatrix} \pi(x_1)\big(1-\pi(x_1)\big) & 0 & \dots & 0 \\ 0 & \pi(x_2)\big(1-\pi(x_2)\big) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \pi(x_n)\big(1-\pi(x_n)\big) \end{bmatrix}$$

2.2 Regresi Logistik Terregularisasi

Permasalahan pada regresi logistik biner yang sering terjadi dalam data berdimensi tinggi adalah terjadinya *overfitting*, yang disebabkan oleh bersarnya variasi estimasi parameter $\boldsymbol{\beta}$. Perlu dilakukan upaya untuk mengatasi permasalahan semacam ini, salah satunya yaitu dengan metode regularisasi. Terdapat dua jenis regularisasi yang dapat diterapkan untuk metode regresi logistik biner, yaitu regularisasi l_1 dan l_2. Regularisasi l_1 menggunakan pinalti dari nilai jumlahan absolut parameter, sedangkan regularisasi l_2 menggunakan pinalti dari nilai jumlahan kuadrat parameter (Ng, 2004).

$$R_1(\boldsymbol{\beta}) = \boldsymbol{\lambda}\|\{\boldsymbol{\beta}\}\|_1 = \boldsymbol{\lambda}\sum_{j=1}^{p}\left|\beta_j\right| \tag{2.10}$$

$$R_2(\boldsymbol{\beta}) = \boldsymbol{\lambda}\|\{\boldsymbol{\beta}\}\|_2 = \boldsymbol{\lambda}\sum_{j=1}^{p}\beta_j^2 \tag{2.11}$$

Persamaan (2.10) menunjukkan regulasi l_1 dan Persaaman (2.11) merupakan regulasi l_2 dengan vector $\boldsymbol{\beta}$ berukuran p x 1 dan memuat parameter $(\beta_1, \beta_2, \ldots, \beta_p)^T$ dan λ merupakan parameter regulasi, dengan $\lambda > 0$. Regularisasi l_1 dapat menghasilkan model yang *sparse*, melakukan seleksi fitur bersaman dengan estimasi variabel. Regulasi l_2 digunakan untuk mengatasi varians estimator yang besar dari hasil estimasi parameter.

Regulasi l_1 dan l_2 pada Persamaan (2.10) dan (2.11) ditambahkan pada fungsi *loss* dari model yang digunakan. Fungsi tujuan metode yang terregularisasi dapat dituliskan sebagai berikut.

$$f(t) = L(t) + R(t) \tag{2.12}$$

dimana,

t = menunjukkan parameter model
f(t) = fungsi tujuan dari metode teregularisasi
L(t) = fungsi *loss* dari metode basis
R(t) = pinalti atau unsur regularisasi

Fungsi *loss* pada metode regresi logistik biner sama dengan negatif dari fungsi ln-*likelihood* pada Persamaan (2.5). Fungsi *loss* $L(\boldsymbol{\beta})$ untuk regresi logistik ditunjukkan sebagai berikut.

$$\ln L(\boldsymbol{\beta}) = -\sum_{i=1}^{n}[y_i \ln \pi(\boldsymbol{x_i}) + (1 - y_i)(\ln(1 - \pi(\boldsymbol{x_i})))] \quad (2.13)$$

dengan membagi fungsi *loss* pada Persamaan (2.13) dengan banyaknya data (n) maka didapatkan persamaan untuk nilai rata-rata fungsi *loss* atau *average loss function* sebagai berikut.

$$L_{avg}(\boldsymbol{\beta}) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \ln \pi(\boldsymbol{x_i}) + (1 - y_i)(\ln(1 - \pi(\boldsymbol{x_i})))] \quad (2.14)$$

Proses estimasi parameter pada metode regresi logistik yang terregularisasi menggunakan *Maximum Likelihood* (MLE) dan metode iterasi *Newton Raphson* seperti Persamaan 2.8 karena hasil dari turunan pertama *MLE* tidak *closed-form*. Estimasi parameter dilakukan dengan meminimalkan *average loss function* yang sudah terboboti dengan pinalti $R(\boldsymbol{\beta})$ seperti persamaan berikut.

$$\min_{\boldsymbol{\beta}}\{L_{avg}(\boldsymbol{\beta}) + R(\boldsymbol{\beta})\} = \min_{\boldsymbol{\beta}}\left\{-\frac{1}{n}\sum_{i=1}^{n} A + R(\boldsymbol{\beta})\right\} \quad (2.15)$$

dimana,

$$A = y_i \ln \pi(\boldsymbol{x_i}) + (1 - y_i)(\ln(1 - \pi(\boldsymbol{x_i})))$$

$R(\boldsymbol{\beta})$ pada Persamaan (2.15) dapat disubtitusi dengan pinalti regularisasi l_1 dan l_2 seperti pada Persamaan (2.10) dan (2.11).

Penelitian ini menggunakan regularisasi l_2 yang selanjutnya akan ditambahkan ke dalam rumusan regresi logisstik biner. Persamaan menggunakan regularisasi l_2 ditunjukkan pada persamaan berikut.

$$\min_{\boldsymbol{\beta}}\{L_{avg}(\boldsymbol{\beta}) + R(\boldsymbol{\beta})\} = \min_{\boldsymbol{\beta}}\left\{-\frac{1}{n}\sum_{i=1}^{n} A + \boldsymbol{\lambda}\sum_{j=1}^{p}\beta_j^2\right\} \quad (2.16)$$

dimana hasil turunan pertama fungsi tujuan regresi logistik terregularisasi sebagai berikut

$$\frac{\partial \ln \min_{\boldsymbol{\beta}}\{L_{avg}(\boldsymbol{\beta})+R(\boldsymbol{\beta})\}}{\partial \boldsymbol{\beta}} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\boldsymbol{x}_i \exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})} - y_i{\boldsymbol{x}_i}^T\right] + 2\boldsymbol{\lambda}\sum_{j=1}^{n}\boldsymbol{\beta}_j$$

$$= \frac{1}{n}\sum_{i=1}^{n}{\boldsymbol{x}_i}^T\left[\frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})} - y_i\right] + 2\boldsymbol{\lambda}\sum_{j=1}^{n}\boldsymbol{\beta}_j$$

$$= \frac{1}{n}\sum_{i=1}^{n}{\boldsymbol{x}_i}^T[\pi(\boldsymbol{x}_i) - y_i] + 2\boldsymbol{\lambda}\sum_{j=1}^{n}\boldsymbol{\beta}_j \qquad (2.17)$$

dengan turunan kedua sebagai berikut

$$\frac{\partial^2 \ln \min_{\boldsymbol{\beta}}\{L_{avg}(\boldsymbol{\beta})+R(\boldsymbol{\beta})\}}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^{\boldsymbol{T}}} = \frac{\partial}{\partial \boldsymbol{\beta}^{\boldsymbol{T}}}\left(\frac{\partial \ln \min_{\boldsymbol{\beta}}\{L_{avg}(\boldsymbol{\beta})+R(\boldsymbol{\beta})\}}{\partial \boldsymbol{\beta}}\right)$$

$$= \frac{\partial}{\partial \boldsymbol{\beta}^{\boldsymbol{T}}}\left(\frac{1}{n}\sum_{i=1}^{n}\left[\frac{\boldsymbol{x}_i \exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})} - y_i{\boldsymbol{x}_i}^T\right] + 2\boldsymbol{\lambda}\sum_{j=1}^{n}\boldsymbol{\beta}_j\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i{\boldsymbol{x}_i}^T\left[\frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})} - \left(\frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}\right)^2\right] + 2\boldsymbol{\lambda}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i{\boldsymbol{x}_i}^T\left(\frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}\right)\left(1 - \frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})}\right) + 2\boldsymbol{\lambda}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i{\boldsymbol{x}_i}^T\,\pi(x_i)\big(1-\pi(x_i)\big) + 2\boldsymbol{\lambda} \qquad (2.18)$$

Besarnya angka pada parameter regularisasi (λ) memiliki pengaruh terhadap nilai estimasi parameter regresi logistik. Nilai Parameter λ yang digunakan pada penelitian ini yang didapatkan dari *tunning parameter* untuk setiap iterasinya yaitu memiliki

range antara 1300 sampai 359000. Parameter λ merupakan pembobot bagi unsur regularisasi yang ditambahkan dalam *average loss function*. Nilai λ yang optimum diperlukan untuk menghasilkan hasil yang baik. Jika nilai λ yang digunakan terlalu besar akan menyebabkan estimasi parameter didominasi oleh unsur regularisasi sehingga model yang dihasilkan memiliki kecenderungan *underfitting*, sebaliknya jika λ semakin mendekati 0, mengakibatkan efek regularisasi yang dihasilkan semakin kecil sehingga *overfitting* dari model yang digunakan tidak dapat diatasi.

2.3 Metode *Ensemble*

Metode *Ensemble* sudah populer dalam *machine learning*. Metode *Ensemble* membangun sejumlah metode klasifikasi dan menggabungkan metode klasifikasi tersebut untuk mendapatkan hasil klasifikasi yang lebih baik. Metode klasifikasi yang digunakan dapat terdiri dari satu jenis metode klasifikasi atau beberapa metode klasifikasi. Metode *Ensembe* sering kali menghasilkan hasil yang lebih baik dibandingkan metode klasifikasi tunggal. Kelebihan dari metode *Ensemble* adalah mampu membuat *weak learners* menghasilkan peningkatan dalam hasil klasifikasinya atau dapat dikatakan *strong learners* (Zhou, 2012).

Terdapat dua jenis algoritma dalam metode *ensemble* yaitu *Bagging* dan *Boosting*. *Bagging* atau *boostrap aggregating* adalah metode untuk memodifikasi hasil algoritma klasifikasi dalam *machine learning*. Metode *Bagging* membuat urutan *classifier* menjadi beberapa urutan. Misalnya terdapat 10 urutan, maka urutan-urutan tersebut sebagai modifikasi dari data *training*. *Classifier* tersebut kemudian digabungkan menjadi satu *classifier* gabungan. Metode *Boosting* mengimprovisasi performa suatu algoritma *learning* untuk membentuk suatu *classifer* yang kuat. Metode Boosting berfokus pada pembuatan model klasifikasi pada setiap iterasi. Data yang akan digunakan bergantung pada hasil klasifikasi pada iterasi sebelumnya. Besarnya nilai *error* menjadi tolok ukur dalam mendapatkan model akhir dalam metode *boosting* guna mendapatkan hasil klasifikasi yang baik (Pangastuti, 2018).

2.4 *Logistic Regression Ensembles* (LORENS)

Tahun 2010, Lim, Ahn, Moon dan Chen mengembangkan metode *Logistic Regression Ensembles* (LORENS) dengan menggunakan *base classifier* yaitu regresi logistik dan berdasarkan algoritma *Logistic Regression Classification By Ensembles From Random Partition* (LR CERP). LORENS mengkombinasikan hasil model regresi logistik untuk mendapatkan satu *classifier* yang kuat dibanding metode agregasi kompleks lainnya guna meningkatkan akurasi prediksi. LORENS menggunakan prosedur yang sama dengan LR CERP, namun prosedur LORENS mengulangi prosedur LR CERP beberapa kali sampai terbentuk beberapa *ensemble*. LORENS mempartisi ruang prediktor Θ yang dipartisi menjadi *s* subruang (θ_1, θ_2,…, θ_s) yang sama. Subruang dipilih secara acak berdasarka distribusi yang sama, diasumsikan tidak terdapat bias pada saat pengambilan prediktor pada masing-masing subruang. Model regresi yang terbentuk pada masing-masing ruang dilakukan tanpa melalui seleksi variabel. Dengan melakukan pengacakan ini, diharapkan probabilitas yang sama pada masing-masing *classifier* pada satu *ensemble* dan juga *error* klasifikasi yang hampir sama.

Peningkatan akurasi dalam satu *ensemble* yang dihasilkan LORENS didapatkan dengan mengkombinasikan nilai prediksi dari model-model regresi logistik pada masing-masing partisi yang didapat. Dengan mengulangi prosedur LR CERP, LORENS mendapatkan kombinasi rata-rata ataupun nilai terbanyak (*majority voting*) yang menghasilkan akurasi yang hampir sama. Rata-rata menghasikan nilai sedikit lebih unggul dari pada nilai terbanyak, sehingga LORENS lebih baik menggunakan nilai rata-rata. Dengan menggunakan prosedur LR CERP, LORENS menghasilkan beberapa *ensemble* dengan partisi acak yang berbeda-beda pula. Dari beberapa *ensemble* yang terbentuk, diambil nilai terbanyak diantaranya. Berdasarkan nilai tersebut dianggap satu akurasi umum. Nilai akurasi tersebut telah ditingkatkan dengan sumbangsih dari beberapa *ensemble* yang dibangun.

LORENS mempunyai kelebihan bebas dari asumsi dimensi data, karena LORENS melakukan partisi secara acak terhadap

prediktornya (Lee, *et al*., 2013). Selain itu kelebihan LORENS terletak dalam penentuan *threshold*. Umumnya *threshold* yang digunakan dalam klasifikasi dengan respon biner adalah 0,5. Apabila proporsi kelas tidak seimbang antara kelas 0 dan kelas 1, klasifikasi yang dihasilkan memiliki akurasi yang tidak akan baik. *Treshold* yang optimal dibutuhkan untuk menyeimbangkan *sensitifity* dan *specifity*. Berikut ini merupakan rumus menghitung *threshold* optimal dari LORENS

$$Treshold = \frac{p + 0{,}5}{2} \tag{2.19}$$

p adalah probabilitas pengamatan yang berada di kelas positif. Berikut ini tahapan dalam proses klasifikasi menggunakan LORENS.

1. Membentuk model logit dari data *training*.
2. Memasukkan data *testing* ke dalam model logit, sehingga diperoleh nlai probabilitas.
3. Mengklasifikasikan pengamatan data *testing*. Jika nilai probabilitasnya lebih besar daripada nilai *threshold* maka pengamatan masuk ke dalam kelas positif, sebaliknya jika nilai probabilitasnya lebih kecil daripada nilai *threshold* maka pengamatan masuk ke dalam kelas negatif.
4. Membandingkan kelas aktual dengan prediksi klasifikasi.
5. Menghitung hasil ketepatan klasifikasi.

2.5 *Ensemble Logistic Regression* (ELR)

Salah satu pengembangan metode regresi logistik biner untuk analisis klasifikasi data berdimensi tinggi adalah metode *Ensemble Logistic Regression* (ELR). Metode ini diperkenalkan oleh Zakharov dan Dupont pada tahun 2011. Metode ini memiliki kemampuan seleksi variabel yang secara bersamaan dilakukan dengan pembuatan model klasifikasi, sehingga dapat dinamakan sebagai metode *embedded*. ELR dibuat dengan menggunakan konsep *ensemble* yang berbasis metode regresi logistik biner terregularisasi (Zakharov & Dupont, 2011).

Metode ELR dibuat dengan berdasarkan regularisasi l_2. Regularisasi l_2 tidak menjamin untuk mendapatkan model yang

sparse dan stabil. Untuk mendapatkan model yang seperti ini, proses pemodelan dilakukan dengan menggunakan *subset* variabel, yaitu sebanyak n dari p variabel. Variabel yang digunakan dalam pemodelan dipilih berdasarkan nilai probabilitas yang telah dihitung untuk setiap variabel. Nilai inisial probabilitas didapatkan dari metode *t-test ranking* seperti yang ditunjukkan pada Persamaan (2.20).

$$t_j = \frac{\mu_{j+} - \mu_{j-}}{\sqrt{\frac{\sigma_{j+}^2}{n_+} + \frac{\sigma_{j-}^2}{n_-}}} \tag{2.20}$$

t_j : Statistik uji *t-test ranking* untuk variabel ke-j, dimana j=1,2,…,p

μ_j : Rata-rata variabel ke-j

σ_j^2 : Varians variabel ke-j

n : Banyaknya sampel

indeks (+) : Kategori 1

indeks (-) : Kategori 0

Inisial vektor probabilitas didapatkan berdasarkan hasil perhitungan 1−*p-value* dari statistik uji pada Persamaan (2.20). Nilai *p-value* ini bisa dihitung atau didapatkan berdasarkan tabel distribusi t dengan derajat bebas sebagai berikut.

$$df = \frac{\sqrt{\frac{\sigma_{j+}^2}{n_+} + \frac{\sigma_{j-}^2}{n_-}}}{\frac{\left(\frac{\sigma_{j+}^2}{n_+}\right)^2}{n_+ - 1} + \frac{\left(\frac{\sigma_{j+}^2}{n_+}\right)^2}{n_- - 1}} \tag{2.21}$$

Hasil dari perhitungan 1−*p-value* hanya digunakan untuk menentukan nilai awal probabilitas variabel, sedangkan nilai probabilitas pada iterasi berikutnya diperbarui sesuai dengan Persamaan (2.22)

$$prob_{j,l} = \frac{1}{z}(prob_{j,l-1} + quality \cdot \beta_j^{2 \cdot sign(quality)}) \tag{2.22}$$

dengan l=1,2,… menunjukkan iterasi yang dilakukan dan nilai inisial probabilitas $prob_{j,0}$ didapatkan berdasarkan nilai t_j. Notasi z menunjukkan normalisasi dari $prob_{j,l}$ dan $quality$ didefinisikan sebagai nilai kualitas relatif yang dapat dihitung sesuai dengan Persamaan (2.23).

$$quality = \log(1 + BCR_l - \overline{BCR}_{l-1}) \tag{2.23}$$

Tabel 2. 1 *Confusion Matrix*

		Kelas Aktual	
		Positif	*Negatif*
Kelas Prediksi	*Positif*	*True Positive (TP)*	*False Positive (FP)*
	Negatif	*False Negative (FN)*	*True Negative (TN)*

Tanda untuk nilai kualitas relatif didapatkan berdasarkan perbandingan nilai $quality$ terhadap nilai BCR_{l-1}, yaitu jika $quality > BCR_{l-1}$, maka ($quality$) adalah positif atau +1, sedangkan jika $quality < BCR_{l-1}$, maka ($quality$) adalah negatif atau −1. BCR atau *Balanced Classification Rate* dihitung berdasarkan Persamaan (2.24).

$$BCR = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{FP + TN}\right) \tag{2.24}$$

TP, TN, FP dan FN didapatkan dari *confusion matrix* yang ditunjukkan seperti pada Tabel 2.1. Perhitungan nilai probabilitas variabel hingga proses mendapatkan nilai $\overline{BCR}_l$ dilakukan secara berulang hingga didapatkan $\overline{BCR}$ yang konvergen. Kriteria konvergensi $\overline{BCR}$ didefinisikan sebagai selisih $\overline{BCR}_l - \overline{BCR}_{l-1}$ atau ε yang sangat kecil. Pembaharuan nilai $\overline{BCR}_l$ menggunakan Persamaan (2.25) sebagai berikut.

$$\overline{BCR}_l = \frac{1}{l+1}(l \cdot \overline{BCR}_{l-1} + BCR_l) \tag{2.25}$$

2.6 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) adalah salah satu turunan dari metode *oversampling*. SMOTE pertama kali diperkenalkan oleh Nithes V. Chawla. Pendekatan ini bekerja dengan membuat replikasi dari data minoritas.

Replikasi tersebut dikenal dengan data sintetis (*syntetic data*). Metode SMOTE bekerja dengan mencari *k nearest neighbors* (yaitu ketetanggaan data) untuk setiap data di kelas minoritas, setelah itu dibuat data sintetis sebanyak persentase duplikasi yang diinginkan dan *k-nearest neighbors* yang dipilih secara acak (Chawla, *et al*., 2002). Persentase duplikasi didaparkan dengan cara menghitung (jumlah observasi data kelas mayor/jumlah observasi data kelas minor) x 100%. *Nearest neighbor* dipilih berdasarkan jarak *Euclidean* antara kedua data, jarak Euclidean (x, y) adalah sebagai berikut.

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \quad (2.26)$$

Secara umum, rumus menentukan data sintetis sebagai berikut.

$$x_{syn} = x_i + (x_{knn} - x_i) \times \delta \quad (2.27)$$

dengan

x_{syn} = data sintetis

x_i = data ke-*i* dari kelas minor yang akan direplikasi

x_{knn} = data dari kelas minor yang memiliki jarak terdekat dengan x_i

δ = bilangan random antara 0 dan 1

Berikut diberikan contoh penggunaan SMOTE

Tabel 2. 2 Contoh Penggunaan SMOTE

No	X_1	X_2	Y
1	2	14	1
2	4	12	1
3	3	19	1
4	6	20	0
5	9	17	0
6	3	19	0
7	6	15	0
8	4	12	0
9	5	11	0

Tabel 2.2 menunjukkan data ilustrasi SMOTE. Variabel respon Y memiliki 2 kelas yaitu kelas 1 dan kelas 0. Kelas minor atau kelas yang paling sedikit adalah kelas 1 (Y=1) berjumlah 3 observasi. Kelas mayor atau kelas yang paling banyak adalah kelas 0 (Y=0) berjumlah 6 observasi. SMOTE digunakan untuk

membuat proporsi kelas variabel respon menjadi seimbang atau *balance*. Langkah pertama adalah menentukan jumlah data sintetis yang akan direplikasi dengan membagi jumlah observasi kelas mayor dengan jumlah observasi kelas minor dikalikan 100% atau dapat dituliskan (6/3 x 100%) dengan hasil sebesar 200%. Sehinggaakan dilakukan 1 replikasi untuk masing-masing observasi dan tetangga data dari data yang akan direplikasi replikasi yaitu dipilih salah satu dari tetangga terdekat atau x_{knn}. Proses membangkitkan data replikasi sebagai berikut. Menentukan tetangga terdekat (x_{knn}) dari data yang akan direplikasi yaitu menghitung jarak *Euclidean* observasi ke-1 dan observasi ke-2 serta observasi ke-1 dan observasi ke-3.
Observasi ke-1 dan observasi ke-2

$$d\left(\begin{bmatrix}2\\14\end{bmatrix}, \begin{bmatrix}4\\12\end{bmatrix}\right) = \sqrt{(2-4)^2 + (14-12)^2} = \sqrt{8} = 2{,}83$$

Observasi ke-1 dan observasi ke-3

$$d\left(\begin{bmatrix}2\\14\end{bmatrix}, \begin{bmatrix}3\\19\end{bmatrix}\right) = \sqrt{(2-3)^2 + (14-19)^2} = \sqrt{26} = 5{,}01$$

Hasil dari perhitungan jarak *Euclidean* diatas, maka diambil jarak tetangga terdekat (x_{knn}) observasi ke-2 yaitu 2,83. Menggunakan Persamaan 2.27 dengan nilai δ sebesar 0,3, data sintesis didapatkan sebagai berikut.

$$x_{syn} = \begin{bmatrix}2\\14\end{bmatrix} + \left(\begin{bmatrix}4\\12\end{bmatrix} - \begin{bmatrix}2\\14\end{bmatrix}\right) \times 0{,}3 = \begin{bmatrix}2{,}6\\13{,}4\end{bmatrix}$$

Berdasarkan hasil perhitungan data sintesis, dapat diketahui data sintesis yang dihasilkan adalah y_{syn}=1, $x_{syn(1)}$ =2,6 dan $x_{syn(2)}$ =13,4. Untuk replikasi dari observasi data minor selanjutnya melalui proses yang sama sampai sebanyak persentase *oversampling* yang ditentukan.

2.7 Evaluasi Ketepatan Klasifikasi

Ukuran dasar yang digunakan mengukur dan mengevaluasi performa klasifikasi adalah sensitivitas, spesifitas dan akurasi. Akurasi menunjukkan efektifitas *classifier* secara menyeluruh. Akurasi biasa digunakan untuk data yang *balanced.* Semakin besar akurasi, maka kinerja *classifier* semakin baik (Okun ,

2011). Berdasarkan Tabel 2.1, berikut perhitungan yang bisa didapatkan.

Akurasi yang digunakan adalah akurasi total dari keseluruhan ketepatan *classifier* dalam mengidentifikasi ke kelas positif maupun negatif. Berikut ini merupakan rumus dari akurasi:

$$Akurasi = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (2.28)$$

TP (*True Positive*) adalah data aktual positif dan diklasifikasikan positif. TN (*True Negatif*) adalah data aktual negatif dan diklasifikasikan negatif. FP (*False Positif*) adalah data aktual negatif namun diklasifikasikan positif. FN (*False Negatif*) adalah data aktual positif namun diklasifikasikan negatif.

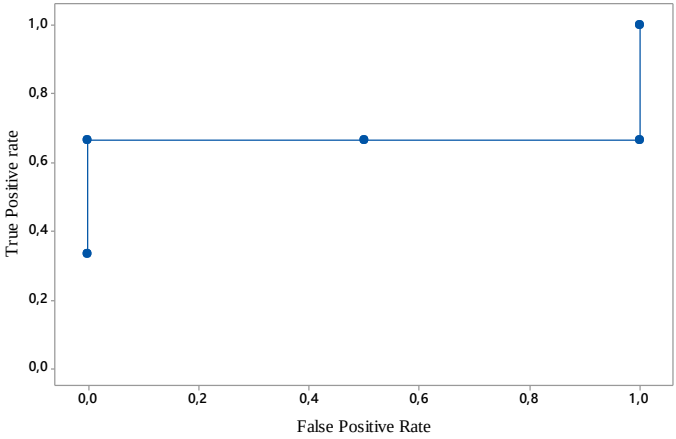
Metode lain dalam mengukur performa klasifikasi adalah menggunakan *Area Under Curve* (AUC). Umumnya, AUC digunakan untuk mengukur klasifikasi apabila data *imbalanced*. Hal ini karena AUC menggunakan sensitivitas dan 1-spesitifitas sebagai dasar pengukuran. Nilai AUC berada diantara 0 dan 1. Apabila nilai AUC semakin mendekati 1, maka model klasifikasi yang terbentuk semakin akurat. Nilai AUC didapatkan dengan menghitung luas area dibawah kurva ROC (Fawcett, 2006).

Berikut ilustrasi penggunaan AUC yang didapatkan dengan menghitung luas daerah di bawah kurva ROC. Perhitungan yang dilakukan ialah perhitungan *True Positif Rate* (TPR) dengan rumus TP/TP+FN serta *False Positive Rate* (FPR) dengan rumus FP/FP+TN. Nilai $\hat{Y}$ akan bernilai 1 jika $p \geq 0.5$ dimana p adalah P(Y=1|X), selain itu maka nilai $\hat{Y}$ dianggap 0.

Tabel 2. 3 Contoh Data Menggunakan AUC

Observasi	Y	$\hat{Y}$	p	TP	FN	FP	TN	TPR	FPR
1	1	1	0,9	1	2	0	2	0,33	0,00
2	1	1	0,7	2	1	0	2	0,67	0,00
3	0	0	0,4	2	1	1	1	0,67	0,50
4	0	0	0,3	2	1	2	0	0,67	1,00
5	1	0	0,1	3	0	2	0	1,00	1,00

Berdasarkan Tabel 2.3 didapatkan nilai AUC seperti pada dengan sebesar 0,6667. Berikut ini akan diberikan ROC *Curve* pada Gambar 2.1



Gambar 2. 1 ROC Curve

2.8 Kanker

Kanker merupakan salah satu penyakit tidak menular yang menjadi masalah kesehatan masyarakat, baik di dunia maupun di Indonesia. Di dunia, 12 persen dari seluruh kematian disebabkan oleh kanker dan merupakan pembunuh nomor 2 setelah penyakit kardiovaskular. Berdasarkan data *GLOBOCAN*, *International Agency for Research on Cancer (IARC)* diketahui bahwa pada tahun 2012 terdapat 14.067.894 kasus baru kanker dan 8.201.575 kematian akibat kanker di seluruh dunia. Jika tidak dikendalikan, diperkirakan 26 juta orang akan menderita kanker dan 17 juta meninggal karena kanker pada tahun 2030 (WHO, 2014). Ironisnya, kejadian ini akan terjadi lebih cepat di negara miskin dan berkembang. *American Cancer Society* (2015), memperkirakan bahwa ada sekitar 1.284.900 kasus kanker baru didiagnosa dan 555.500 kematian akibat kanker terjadi di USA tahun 2002 lalu. Di Indonesia dan dunia tiap tahun kasus kanker terus meningkat. Mulai dari yang tertinggi seperti kanker payudara, kanker leher rahim (*serviks*), kanker paru, kanker usus besar (kolorektal), kanker prostat, kanker darah,kanker tulang,

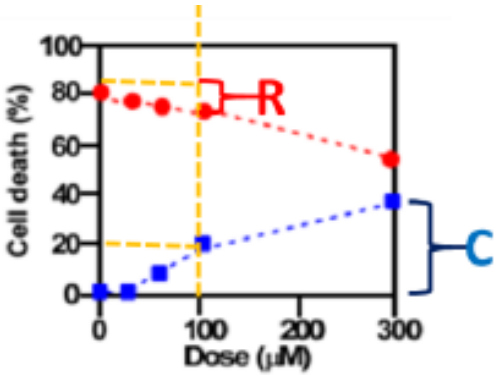
kanker hati, dan kanker kulit. Setidaknya di dunia ada lebih dari 100 jenis kanker.

Radiasi merupakan modalitas yang sangat mapan (*well-established*) untuk pengobatan berbagai macam kanker dimana hampir setengah dari pasien kanker menjalani radioterapi untuk pengobatan penyakitnya. Akan tetapi, keampuhan pengobatan dengan radiasi masih terkendala bahwa dosis yang diberikan ke sel kanker dapat merusak jaringan normal di daerah yang terkena radiasi. Toksisitas berkaitan dengan radiasi dapat menjadi serius dan terkadang mengancam kehidupan pasien (melemahkan secara serius). Sifat toksisitasnya bervariasi bergantung pada jenis partikel, energi, intensitas, skema fraksionasi dan dosis totalnya. Di samping itu, lokasi kanker, volume total yang diradiasi, dan kondisi kesehatan pasien dapat mempengaruhi efek radiasi pada pasien. Beberapa efek samping meliputi leukopenia, anemia, nausea (muntah), diare, reaksi kulit, alopecia (rambut rontok), mukositis, gangguan kognitif dan saraf lain, pneumonitis, fibrosis dan munculnya neoplasma kedua. Jadi, diperlukan adanya strategis inovatif yang dapat memperkecil volume dan keparahan kerusakan jaringan normal tanpa meminimalkan keuntungan efek anti-tumor dari radioterapi (Nurhayati, *et al.*, 2011).

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Sumber data yang digunakan dalam penelitian ini merupakan data sekunder dari penelitian Ariyasu, *et al*. (2014) yang selanjutnya digunakan oleh Matsumoto, *et al.* (2016) dengan data yang telah *pre-processing* sedemikian rupa yaitu data penyusun senyawa yang berhubungan dengan p53 inhibitor. Data terdiri dari 217 variabel prediktor dengan 84 observasi. Variabel prediktor didapatkan menggunakan *Discovery Studio* yang merupakan *software* pemodelan 3D yang dapat menghitung *chemical properties*. Dalam menentukan kelas (1 atau 0) pada variabel respon digunakan tingkat kematian sel sebagai indikator dengan *treshold* yang berbeda. Toksisitas menggunakan tingkat kematian sel pada sel normal dan proteksi radiasi menggunakan tingkat kematian sel pada sel kanker. Penentuan *treshold* didapatkan dari hasil penelitian Ariyasu, *et al*(2014). Ilustrasi penentuan *treshold* dapat dilihat pada Gambar 3.1 berikut. Nilai C merupakan *threshold* untuk toksisitas dan nilai R *threshold* proteksi radiasi. *Treshold* yang didapatkan untuk toksisitas adalah sebesar 20% dan proteksi radiasi sebesar 10%.



Gambar 3. 1 Penentuan *Threshold* untuk Toksisitas dan Proteksi Radiasi

3.2 Variabel Penelitian

Variabel yang digunakan dalam penelitian ini terdiri dari 217 variabel prediktor (zat penyusun senyawa) dan 84 senyawa dengan dua variabel respon yaitu Y_1 dan Y_2, dimana Y_1 adalah fungsi proteksi radiasi dengan perbandingan kelas 1 dan 0 yaitu 39:45, serta Y_2 adalah toksisitas yang memiliki perbandingan kelas 1 dan 0 sebesar 42:42. Dalam menentukan kelas variabel respon baru dilakukan skema sebagai berikut. Variabel Y_1 untuk proteksi radiasi dengan kategori kelas 0 yaitu proteksi radiasi rendah dimana tingkat kematian sel kanker kurang dari sama dengan 10% dan kategori kelas 1 yaitu proteksi radiasi tinggi dimana tingkat kematian sel kanker lebih dari 10%, serta variabel Y_2 untuk toksisitas dengan kategori kelas 0 yaitu toksisitas dengan tingkat kematian sel normal lebih dari 20% dan kategori kelas 1 yaitu toksisitas dengan tingkat kematian sel normal kurang dari sama dengan 20%. Berdasarkan variabel respon Y_1 dan Y_2 akan dibentuk variabel respon baru dengan kriteria kategori kelas 1 yaitu proteksi radiasi dengan tingkat kematian sel kanker tinggi atau lebih dari 10% dan toksisitas dengan tingkat kematian sel normal rendah atau kurang dari 20%. Kriteria variabel respon baru dengan kategori kelas 0 yaitu memiliki kriteria tingkat kematian sel kanker pada proteksi radiasi tinggi dan sel normal toksisitas tinggi, tingkat kematian sel kanker proteksi radiasi rendah dan sel normal toksisitas rendah dan kriteria terakhir yaitu tingkat kematian sel kanker proteksi radiasi rendah dan sel normal toksisitas tinggi. Variabel respon baru yang terbentuk menghasilkan data *imbalance* dengan rasio kelas 1 dan kelas 0 yaitu 9:75, dimana kelas 1 merupakan kelas minor dan kelas 0 termasuk kelas mayor. Variabel penelitian dengan variabel respon baru akan ditampilkan pada Tabel 3.1.

Tabel 3. 1 Variabel Penelitian

Variabel	Nama Variabel	Skala
Respon (Y)	Target kelas $Y(1)$ = Proteksi radiasi tinggi dengan tingkat kematian sel kanker > 10% (Y_1>10%) dan Toksisitas dengan tingkat kematian sel normal < 20% (Y_2<20%) $Y(0)$ = Selainnya	Nominal
Prediktor (x_i)	x_1 = pKa (tingkat keasaman)	Rasio
	x_2 = Jumlah atom Br (Br_Count)	Rasio
	⋮	
	x_{216} = Jumlah molekul 3D SAVol (Molecular_3D_SAVoL)	Rasio
	x_{217} = Volume molekul (Molecular_Volume)	Rasio

Struktur data yang digunakan pada penelitian ini sebagai berikut.

Tabel 3. 2 Struktur Data

No	Senyawa	Variabel Respon (Y)	Variabel Prediktor (X)				
			X_1	X_2	X_3	…	X_{217}
1	AS-1	Y_1	$X_{1,1}$	$X_{2,1}$	$X_{3,1}$	…	$X_{217,1}$
2	AS-10	Y_2	$X_{1,2}$	$X_{2,2}$	$X_{3,2}$	…	$X_{217,2}$
3	AS-11	Y_3	$X_{1,3}$	$X_{2,3}$	$X_{3,3}$	…	$X_{217,3}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
84	YT-1	Y_{84}	$X_{1,84}$	$X_{2,84}$	$X_{3,84}$	…	$X_{217,84}$

3.3 Langkah Analisis

Langkah analisis yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Melakukan eksplorasi data menggunakan statistika deskriptif.
2. Melakukan penanganan *imbalance* data *training* dengan *Synthetic Minority Oversampling Technique* (SMOTE). Langkah menggunakan SMOTE sebagai berikut.
   a. Menentukan jumlah data kelas mayor (Y_0) dan kelas minor (Y_1).
   b. Menentukan presentase data SMOTE yang akan digunakan dengan rumus (jumlah data kelas mayor/jumlah data kelas minor) x 100%.
   c. Menentukan data *k-Nearest Neighbour* (x_{knn}) berjumlah 8 yaitu jarak terdekat dari setiap data minor yang akan disintesis menggunakan persamaan (2.27).
3. Melakukan klasifikasi menggunakan metode *Logistic Regression Ensembles* (LORENS) dengan SMOTE dan tanpa penanganan data *imbalance* melalui langkah-langkah berikut.
   a. Menentukan pembagian data *training* dan *testing* dengan proporsi 80:20 secara *stratified* berdasarkan kelas variabel respon.
   b. Menentukan banyak subruang (*s*) dimana *s* = 5, 8, 10, 15, 20, 25, 30, 40, 45, dan 50.
   c. Menentukan banyaknya *ensemble* (*m*) dimana *m* = 11.
   d. Menentukan nilai *threshold* sebesar 0,5 dan menghitung nilai *threshold* optimum dengan Persamaan (2.19).
   e. Membuat subruang (*s*) yang merupakan partisi dari variabel prediktor yang dipilih secara *random sampling* dari data *training* untuk satu *ensemble*.
   f. Membuat model regresi logistik menggunakan data *training* dari masing-masing partisi subruang (*s*).
   g. Memprediksi data *testing* menggunakan model regresi logistik dari data *training*.
   h. Menghitung rata-rata probabilitas masing-masing model dari setiap partisi subruang.
   i. Mengklasifikasikan hasil rata-rata probabilitas berdasarkan *threshold* 0,5 dan *threshold* optimum dimana jika rata-rata probabilitas bernilai lebih dari nilai

treshold maka diklasifikasikan ke kelas positif atau 1 dan jika kurang dari nilai *treshold* masuk ke dalam klasifikasi kelas negatif atau 0.

j. Mengulangi langkah v sampai ix hingga terbentuk *m ensemble*.

k. Mencari nilai prediksi terbanyak (*majority voting*) untuk masing-masing observasi diantara semua *ensemble* dari *threshold* 0,5 dan *threshold* optimum.

4. Melakukan klasifikasi menggunakan metode *Ensemble Logistic Regression* (ELR) dengan SMOTE dan tanpa penanganan data *imbalance* dengan langkah sebagai berikut.

a. Membagi data *training* dan data *testing* secara *stratified* dengan proporsi 80% data *training* dan 20% data *testing*

b. Menentukan nilai parameter teregulasi λ dan nilai $\overline{BCR}_0$ yang merupakan rata-rata probabilitas kelas positif.

c. Menentukan inisial vektor probabilitas awal menggunakan nilai 1-*pvalue* dari *t test ranking*.

d. Mendapatkan variabel prediktor terpilih yang akan digunakan berdasarkan nilai inisial vektor probabilitas awal.

e. Membuat model regresi logistik teregularisasi dengan variabel prediktor yang terpilih.

f. Memprediksi data *training* menggunakan model yang didapatkan dari data *training*. Prediksi kelas bernilai 1 apabila nilai probabilitas lebih dari 0,5 dan kelas 0 apabila nilai probabilitas kurang dari 0,5.

g. Menghitung nilai BCR_l dari hasil prediksi yang telah didapatkan.

h. Mendapatkan inisial probabilitas baru untuk variabel terpilih menggunakan Persamaan 2.22.

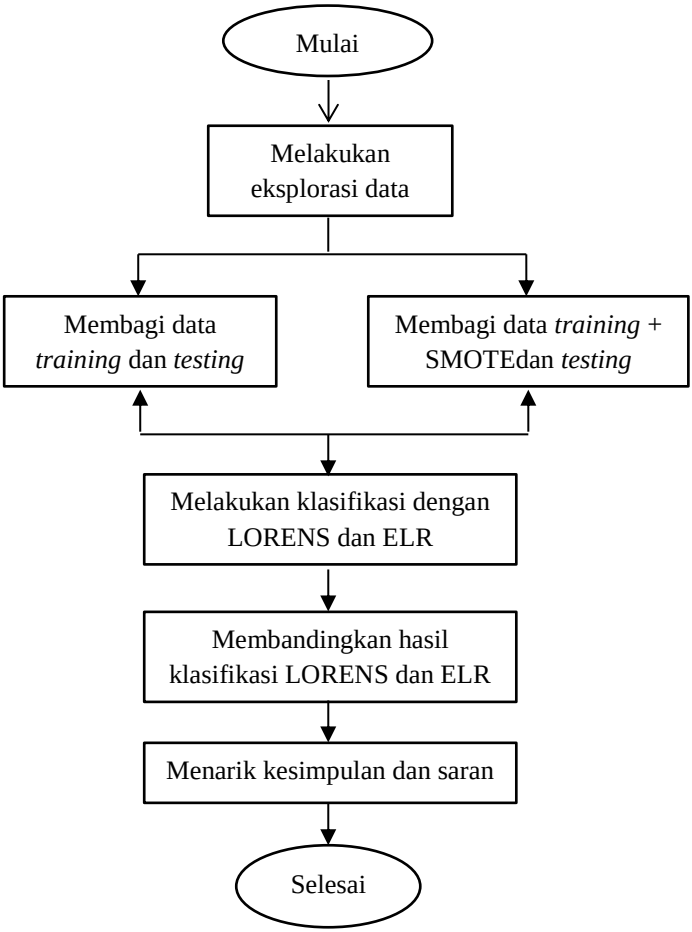
i. Menghitung nilai $\overline{BCR}_1$.

j. Mendapatkan nilai ε. Jika nilai ε lebih dari 10^{-5} maka langkah (e) sampai langkah (i) diulang hingga mendapat nilai ε kurang dari 10^{-5} dengan nilai inisial vektor probabilitas baru untuk setiap iterasi. Model terbaik telah didapatkan apabila nilai ε telah kurang dari 10^{-5}.

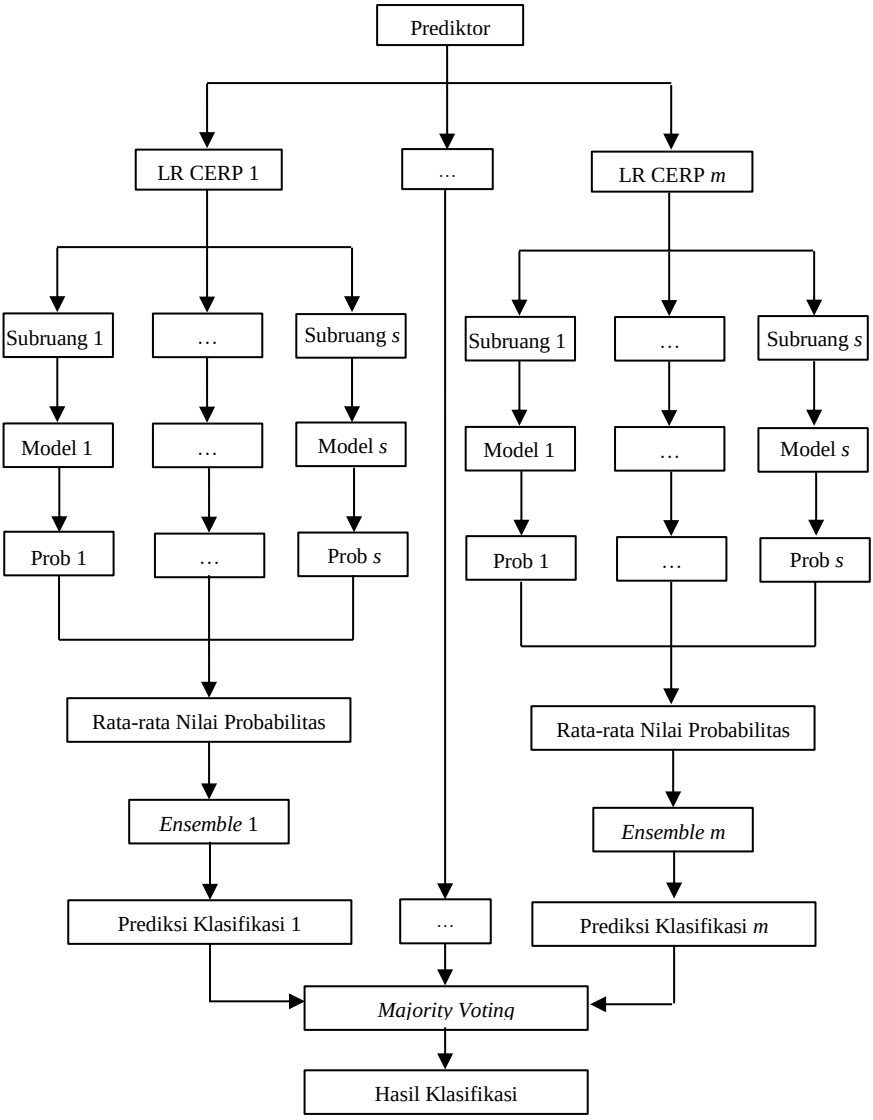
k. Mengklasifikasikan data menggunakan model terbak dan mendapatkan hasil evaluasi ketepatan klasifikasi.

5. Membandingkan hasil klasifikasi menggunakan metode *Logistic Regression Ensembles* (LORENS) dan *Ensemble Logistic Regression* (ELR) dengan SMOTE dan tanpa penanganan data *imbalance*.

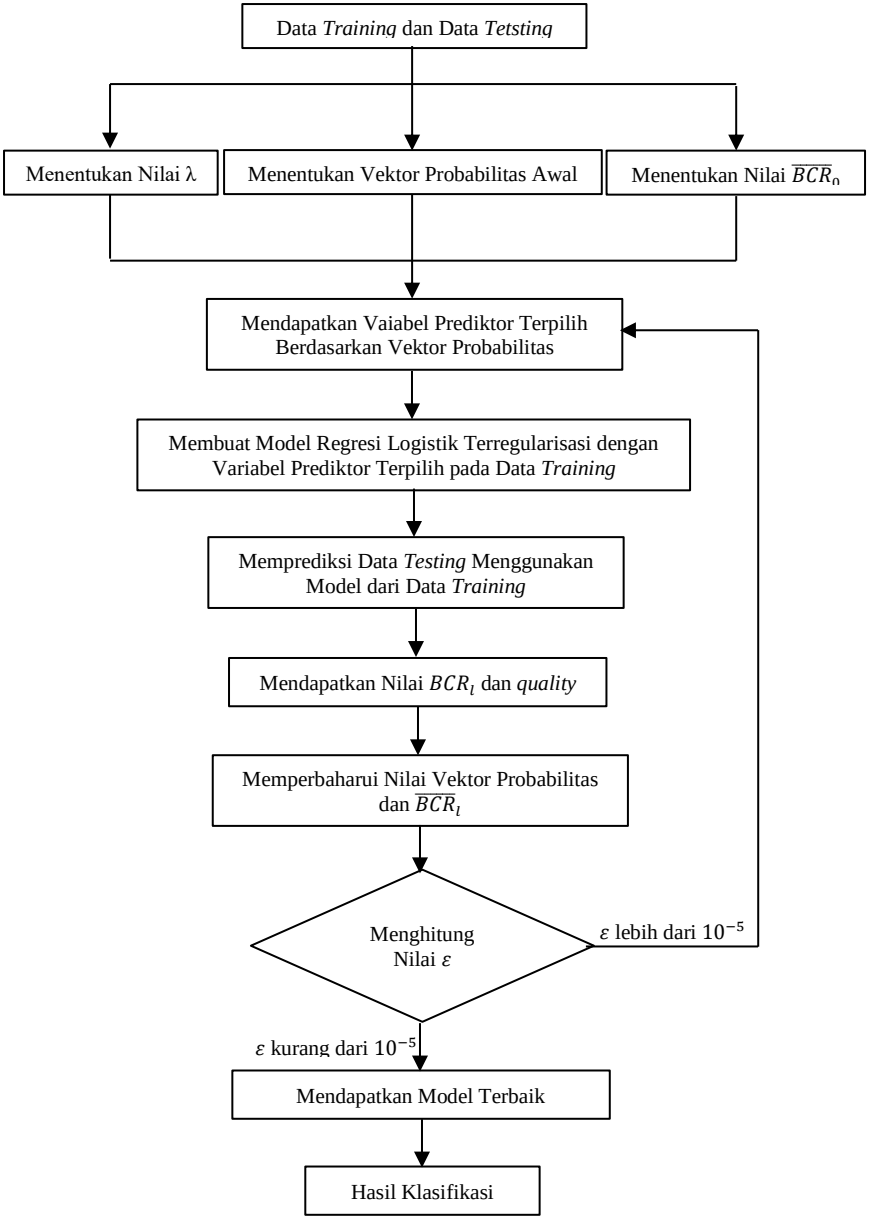
Adapun diagram alir untuk langkah-langkah penelitian yang diberikan pada Gambar 3,2 sebagai berikut.



Gambar 3. 2 Diagram Alir



Gambar 3. 3 Bagan Konsep LORENS



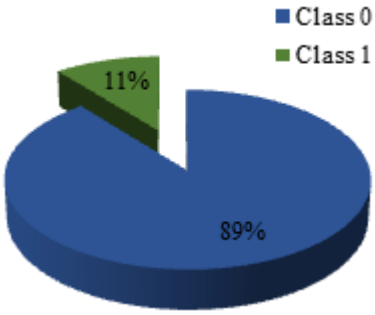
Gambar 3. 4 Bagan Konsep ELR

BAB IV
ANALISIS DAN PEMBAHASAN

Pada penelitian ini akan dilakukan klasfikasi senyawa obat kanker yang dianggap baik sebagai *radiopretector* menggunakan metode *Logistic Regression Ensemble* (LORENS) dan *Ensemble Logistic Regression* (ELR) dengan penanganan data *imbalance* menggunakan *Synthetic Minority Oversampling Technique* (SMOTE) terhadap masing-masing metode serta akan dibandingkan hasil klasifikasi dari metode yang digunakan. Ukuran yang digunakan dalam membandingkan hasil klasifikasi kedua metode yaitu akurasi dan *Area Under Curve* (AUC).

4.1 Karakteristik Senyawa Obat Kanker

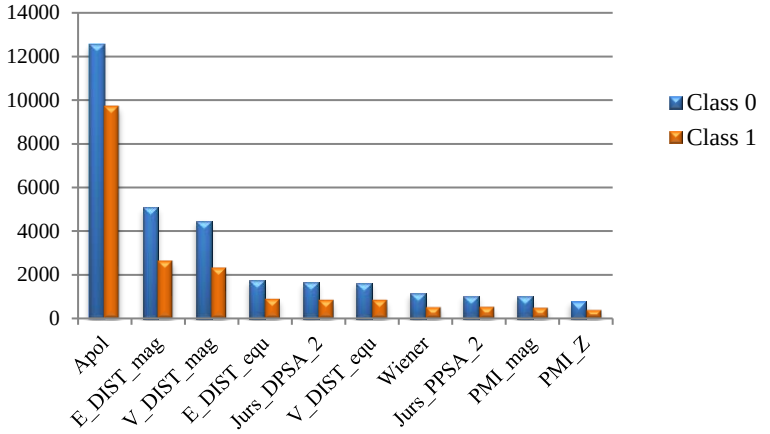
Karakteristik data sangat penting untuk diketahui terlebih dahulu sebelum melakukan langkah lebih lanjut dalam mengolah data. Pada bagian ini akan dilakukan analisis eksplorasi data untuk mengetahui karakteristik data senyawa obat kanker. Variabel respon yang digunakan merupakan variabel respon baru yang telah dihasilkan menggunakan skema pada Sub Bab 3.2.



Gambar 4. 1 Proporsi Kelas Senyawa Obat Kanker

Variabel respon baru yang terbentuk memiliki rasio 9:75, dimana rasio tersebut menunjukkan bahwa 9 observasi masuk kedalam kelas 1 atau dianggap baik sebagai *radioprotector* dan 75 observasi lainnya masuk kedalam kelas 0 atau tidak dianggap baik sebagai *radioprotector*. Pada Gambar 4.1 dapat diketahui bahwa rasio dari kedua kelas memiliki proporsi dengan

perbandingan 11%:89%, hal ini menunjukkan bahwa dari variabel respon baru yang telah terbentuk terjadi kasus *imbalance data*. Data yang *imbalance* dapat mempengaruhi hasil prediksi dari klasifikasi dan hasil tersebut akan cenderung mengikuti kelas terbanyak atau mayor, sehingga diperlukan penanganan terlebih dahulu pada kasus kelas yang *imbalance* tersebut.



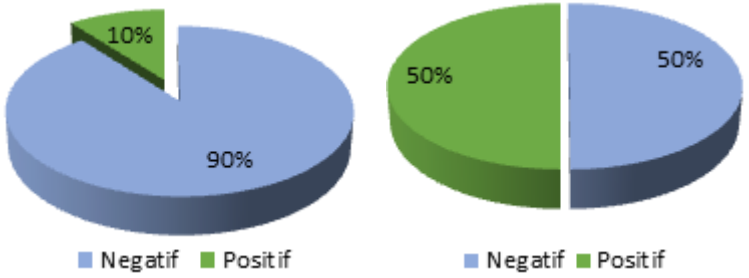
Gambar 4. 2 Perbandingan Rata-rata Variabel Prediktor Antar Kelas

Gambar 4.2 menunjukkan terdapat perbedaan nilai rata-rata antar kelas positif dan kelas negatif yang digunakan. Hal ini ditunjukkan dari nilai rata-rata variabel prediktor Apol kelas negatif memiliki rata-rata lebih besar dibandingkan kelas positif. Begitu juga dengan Variabel E_DIST_mag memiliki nilai rata-rata kelas negatif lebih besar dibandingkan kelas positif sampai Variabel PMI_Z dimana kelas negatif nilai rata-ratanya lebih besar dibanding kelas positif.

4.2 Analisis *Synthetic Minority Oversampling Technique* (SMOTE) Senyawa Obat Kanker Untuk Optimasi Proteksi Radiasi dan Toksisitas

Variabel respon data klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas memilili proporsi kelas yang tidak seimbang atau *imbalance* yaitu 9 positif dan 75 negatif. Data kemudian dibagi menjadi data training dan data

testing dengan ukuran 80% data *training* dan 20% data testing secara *stratified* atau memperhatikan proporsi kelas dalam pembagian data. Proporsi kelas pada data *training* yaitu 7 positif dan 60 negatif. Proporsi kelas data *testing* yaitu 2 positif dan 15 negatif. Untuk mengatasi data yang *imbalance* menggunakan *Synthetic Minority Oversampling Technique* (SMOTE). SMOTE dilakukan hanya pada data *training*. Jika SMOTE dilakukan pada data *training* dan data *testing* hasil yang didapatkan akan terlalu baik atau *overoptimistic* karena data *testing* mengalami replikasi dengan pola data yang sama pada data *training* dan data *testing*.



(a) **(b)**

Gambar 4. 3 (a) Data *Training Imbalance* , (b) Data *Training Balance*

Berdasarkan Gambar 4.3 dapat diketahui bahwa proporsi kelas *imbalance* data *training* senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas kelas positif dengan kelas negatif sebesar 10% : 90% atau 7:60. Setelah digunakan metode SMOTE untuk menyeimbangkan kelas, proporsi kelas data *training* menjadi *balance* sebesar 50%:50% atau kelas positif dan kelas negatif berjumlah sama yaitu 60 untuk masing-masing kelas.

4.3 Klasifikasi Senyawa Obat Kanker Untuk Optimasi Proteksi Radiasi dan Toksisitas Menggunakan Metode *Logistic Regression Ensemble* (LORENS)

Pada bagian ini akan dilakukan klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode LORENS. Data yang digunakan dalam penelitian ini merupakan data *imbalance* sehingga digunakan hasil analisis SMOTE pada Sub Bab 4.2 untuk mengatasi

imbalance. Penggunaan SMOTE diharapkan mampu meningkatkan evaluasi ketepatan klasifikasi yaitu akurasi dan AUC.

4.3.1 Analisis *Logistic Regression Ensemble* (LORENS)

Logistic Regression Ensemble atau LORENS merupakan metode yang berbasis pada *Logistic Regression*. LORENS menggunakan konsep dari metode LR CERP dengan mengulanginya hingga terbentuk *m ensemble*. LORENS tidak memerlukan asumsi apapun dalam pengaplikasiannya. Klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas dengan metode LORENS menggunakan data dengan variabel prediktor berjumlah 217 dan jumlah observasi sebanyak 84. Data yang digunakan dibagi menjadi 2 yaitu *training* dan *testing* dengan proporsi 80% data *training* dan *20%* data *testing* secara *stratified*. Data yang digunakan termasuk *high dimensional* yaitu jumlah variabel lebih banyak dibandingkan jumlah observasi. Metode LORENS dapat menangani *high dimensional* dengan membagi variabel ke dalam subruang *s* dengan jumlah yang telah ditentukan sehingga data yang digunakan tidak termasuk *high dimensional*. Jumlah subruang minimal yang dapat digunakan yaitu dimisalkan *v* adalah jumlah variabel dan *n* adalah jumlah observasi data *training* maka jumlah subruang minimal adalah $\frac{v}{n-1} = \frac{217}{67-1}$= 3,288 ≈ 4 dengan jumlah variabel untuk tiga subruang pertama yaitu 54 variabel dan subruang terakhir terdapat 55 variabel. Jumlah subruang maksimal yang dapat digunakan adalah sejumlah banyaknya *v* variabel atau 217 variabel. Jumlah maksimal variabel yang dapat digunakan dalam satu subruang adalah berjumlah *n*-1 = 67-1 = 66 variabel.

Penelitian ini menggunakan subruang *s* sebanyak 10 subruang yaitu 5, 8, 10, 15, 20, 25, 30, 40, 45, dan 50. Masing-masing subruang akan berisi beberapa variabel prediktor. Skenarionya dapat diilustrasikan sebagai berikut. Menggunakan subruang berjumlah 5 akan terdapat sebanyak 43 variabel prediktor untuk subruang urutan ke-1, ke-2 dan ke-3 serta 44 variabel prediktor untuk subruang urutan ke-4 dan ke-5. *Ensemble* yang digunakan berjumlah 11 yaitu untuk menghindari jumlah

yang seri dalam *majority voting*. *Treshold* yang digunakan sebesar 0,5 dan *threshold* optimum yang didapatkan menggunakan Persamaan 2.16 sebesar 0.3022. Kedua *threshold* akan dibandingkan menurut evaluasi ketepatan klasifikasi yang terbaik.

Analisis metode LORENS dengan 5 subruang dan *threshold* 0,5 akan melalui tahapan seperti yang telah dijelaskan pada Sub Bab 3.3. Berikut ini diberikan tabel partisi data *training* dengan variabel prediktor yang dipilih secara *random sampling* untuk tiap subruang dalam sebelas *ensemble* yang digunakan.

Tabel 4. 1 Partisi Variabel Prediktor 5 Subruang Treshold 0.5

Nomor	Variabel Prediktor	*Ensemble*										
		1	2	3	4	5	6	7	8	9	10	11
1	pKa.max20.	4	2	1	1	2	3	3	4	5	1	1
2	Br_Count	1	3	1	2	2	4	3	2	4	4	1
3	C_Count	1	3	2	3	5	3	2	2	1	3	1
4	Cl_Count	3	3	5	3	4	5	3	4	5	5	3
5	F_Count	3	1	3	2	3	4	5	5	2	3	3
	⋮						⋮					⋮
217	Molecular_ Volume	2	5	2	3	2	2	4	3	3	2	1

Tabel 4.1 menunjukkan hasil dari partisi variabel prediktor untuk *ensemble* pertama dan dapat diperhatikan bahwa variabel pKa.max.20 masuk kedalam subruang keempat, BR_Count bertempat pada subruang pertama, C_Count terdapat pada subruang pertama, CI_Count masuk ke subruang ketiga, F_Count terdapat di subruang ketiga hingga seterusnya sampai variabel terakhir yaitu Molecular_Volume yang termasuk ke dalam subruang kedua. *Ensemble* selanjutnya yaitu *ensemble* kedua hingga kesebelas penempatan variabel prediktor secara *random sampling* dipartisi dan masuk ke dalam subruang mengikuti skema seperti pada *ensemble* pertama.

Tabel 4. 2 Intercept Persamaan Regresi 5 Subruang *Treshold* 0.5

Subruang	*Ensemble*			
	1	2	…	11
1	-1200,9643	2578,5412		1892,5693
2	1453,5087	-1896,5820		273,0556
3	-174,7352	-1402,1170	…	-5134,6839
4	-7287,1864	56,3230		-167,2830
5	1833,0864	-1963,3962	…	0,4741

Tabel 4. 3 Koefisien Regresi 5 Subruang Threshold 0.5

No	Variabel	*Ensemble*			
		1	2	…	11
1	pKa.max20	4,8242	-1,6320		-2,0914
2	Br_Count	14,6884	308,6590		0,8149
3	C_Count	386,9823	86,0021	…	105,3888
4	Cl_Count	-103,5001	59,4716		-200,8560
5	F_Count	-122,3491	43,2922		-94,9749
6	H_Count	-471,7253	120,7388		31,4449
7	I_Count	-870,4214	-17,6324		5019,5330
8	N_Count	289,4989	-186,5455	…	-69,2935
9	O_Count	145,0979	-189,8745		-0,4375
⋮	⋮	⋮	⋮	⋱	⋮
216	Molecular_ 3D_SAVol	-2,0475	-2,9214		-5,3840
217	Molecular _Volume	-3,7283	3,8323	…	-2,2657

Catatan:

Subruang ke-1 Berwarna	
Subruang ke-2 Berwarna	
Subruang ke-3 Berwarna	
Subruang ke-4 Berwarna	
Subruang ke-5 Berwarna	

Setelah didapatkan partisi variabel prediktor dengan 5 subruang pada masing-masing *ensemble*, selanjutnya membuat model *Logistic Regression* menggunakan data *training*. *Ensemble* kesatu sampai *ensemble* kesebelas akan membentuk masing-masing 5 model *Logistic Regression*. Nilai koefisien regresi yang terbentuk ditunjukkan pada Tabel 4.2 dan Tabel 4.3.

Nilai koefisien regresi pada Tabel 4.2 dan Tabel 4.3 disubtitusikan ke dalam Persamaan 2.2 sehingga terbentuk 5 model regresi. Model regresi yang telah terbentuk dari data *training* digunakan untuk mendapatkan nilai probabilitas sejumlah *n* observasi data *testing* pada masing-masing subruang untuk setiap *ensemble* sehingga akan terbentuk lima probabilitas setiap observasi data *testing*. Probabilitas yang dihasilkan berguna untuk memprediksi kelas pada data testing. Kelima probabilitas yang telah didapatkan dirata-rata guna medapatkan probabilitas akhir satu *ensemble* sehingga akan terdapat rata-rata probabilitas sebanyak *n* observasi data *testing* dan sebelas *ensemble*.

Tabel 4. 4 Hasil Akhir Probabilitas LORENS 5 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0,000601	0,2	3,06E-06	⋯	0,2
2	7,16E-34	1,15E-12	3,08E-12		2,16E-23
3	4,69E-17	2,67E-11	2,14E-21		5,85E-23
4	3,58E-14	0,2	2,68E-12		4,15E-05
5	0,200082	0,2	1,67E-21		0,6
6	0,193321	0,276262	0,618253		0,386402
7	0,2	2,38E-32	8,39E-07		0,2
8	0,8	0,600001	0,795526		0,4
9	1,28E-38	0,00534	4,55E-63	⋯	0,2
10	8,11E-22	0,199465	1,22E-52		8,94E-23
11	0,000172	0,400735	0,2		0,2
12	0,04801	0,200021	0,6		0,4
13	0,401959	0,399999	0,400482		0,6

Tabel 4.4 Hasil Akhir Probabilitas LORENS 5 Subruangg *Threshold* 0,5 (Lanjutan)

No	*Ensemble*				
	1	2	3		11
14	0,215786	0,4	0,4		0,2
15	0,6	0,6	0,6		0,225894
16	0,399983	0,395689	0,4		0,270724
17	0,400271	0,400111	0,6	⋯	0,2

Hasil akhir probabilitas LORENS 5 subruang *threshold* 0,5 ditunjukkan pada Tabel 4.4. Apabila nilai probabilitas akhir lebih besar dari nilai *threshold* maka pengamatan akan masuk ke dalam kelas 1 dan apabila nilai probabilitas akhir lebih kecil dari nilai *threshold* maka pengamatan masuk kelas 0. Hasil klasifikasi LORENS didapatkan dengan memprediksi kelas data *testing* yaitu *majority voting* hasil kelas nilai probabilitas akhir.

Tabel 4. 5 Hasil Klasifikasi LORENS Majority Voting 5 Subruang *Threshold* 0,5

No	*Ensemble*											Vote		Prediksi
	1	2	3	4	5	6	7	8	9	10	11	0	1	
1	0	0	0	0	0	0	0	0	0	0	0	11	0	0
2	0	0	0	0	0	0	0	0	0	0	0	11	0	0
3	0	0	0	0	0	0	0	0	0	0	0	11	0	0
4	0	0	0	0	0	1	0	0	0	0	0	10	1	0
5	0	0	0	0	0	0	0	0	0	0	1	10	1	0
6	0	0	1	1	0	1	0	0	1	0	0	7	4	0
7	0	0	0	0	0	0	0	0	1	0	0	10	1	0
8	1	1	1	0	1	1	0	0	0	0	0	6	5	0
9	0	0	0	0	0	0	0	0	0	0	0	11	0	0
10	0	0	0	0	0	0	0	0	0	0	0	11	0	0
11	0	0	0	0	0	0	0	1	0	0	0	10	1	0
12	0	0	1	1	0	0	1	1	1	0	0	6	5	0
13	0	0	0	1	1	1	1	0	0	0	1	6	5	0

Tabel 4.5 Hasil Klasifikasi LORENS Majority Voting 5 Subruang Threshold 0,5 (Lanjutan)

No	*Ensemble*											Vote		Prediksi
	1	2	3	4	5	6	7	8	9	10	11	0	1	
14	0	0	0	0	0	0	1	0	0	0	0	10	1	0
15	1	1	1	1	1	1	1	1	0	1	0	2	9	1
16	0	0	0	0	0	0	0	0	0	0	0	11	0	0
17	0	0	1	0	1	0	1	1	1	1	0	5	6	1

Berdasarkan Tabel 4.5 dapat diperhatikan pada observasi pertama mendapatkan vote kelas 0 sebanyak 11 dan kelas 1 sejumlah 0 dari kesebelas *ensemble*. Selanjutnya dilakukan *majority voting* untuk observasi pertama. Hasil *majority voting* menunjukkan kelas 0 memiliki jumlah yang lebih banyak, maka prediksi kelas data *testing* observasi pertama yaitu kelas 0. Observasi kedua memiliki jumlah vote kelas 0 lebih banyak dari kelas 1 yaitu 11 dan 0. Hasil prediksi observasi data *testing* observasi kedua yaitu kelas 0. Observasi selanjutnya dilakukan dengan proses yang sama hingga observasi ketujuh belas dimana perhitungan vote kelas 0 memiliki jumlah 5 dan kelas 1 berjumlah 6. Sehingga hasil prediksi kelas data *testing* observasi ketujuh belas yaitu kelas 1.

Hasil klasifikasi partisi subruang berukuran 8, 10, 15, 20, 25, 30, 40, 45 dan 50 dengan *threshold* 0,5 dan *threshold* optimum selanjutnya mengikuti proses seperti subruang 0,5 hingga memperoleh prediksi akhir hasil *majority voting*. Hasil *confusion matrix* klasifikasi seluruh subruang menggunakan *threshold* 0,5 dan *threshold* optimum ditunjukkan pada tabel dibawah ini.

Tabel 4. 6 *Confusion Matrix* LORENS

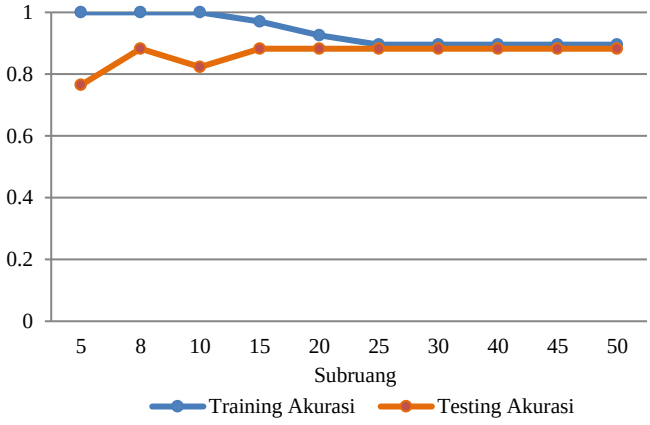
			Kelas Aktual			
			Threshold 0,5		*Threshold* Optimum	
			Positif	Negatif	Positif	Negatif
Kelas Prediksi	5 Subruang	Positif	0	2	0	7
		Negatif	2	13	2	8

Tabel 4.6 *Confusion Matrix LORENS* (Lanjutan)

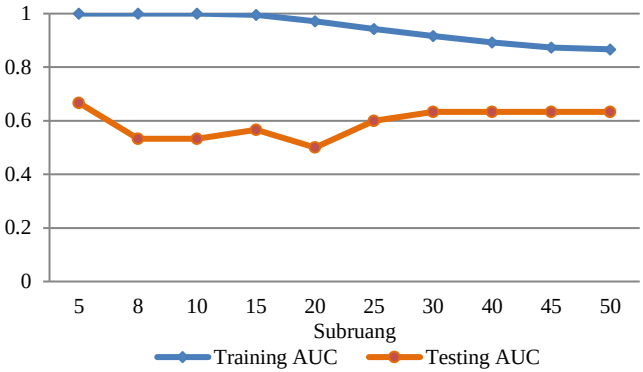
			Kelas Aktual			
			Threshold 0,5		*Threshold* Optimum	
			Positif	Negatif	Positif	Negatif
Kelas Prediksi	8 Subruang	Positif	0	0	0	5
		Negatif	2	15	2	10
	10 Subruang	Positif	0	1	0	3
		Negatif	2	14	2	12
	15 Subruang	Positif	0	0	0	3
		Negatif	2	15	2	12
	20 Subruang	Positif	0	0	0	1
		Negatif	2	15	2	14
	25 Subruang	Positif	0	0	0	0
		Negatif	2	15	2	15
	30 Subruang	Positif	0	0	0	0
		Negatif	2	15	2	15
	40 Subruang	Positif	0	0	0	0
		Negatif	2	15	2	15
	45 Subruang	Positif	0	0	0	0
		Negatif	2	15	2	15
	50 Subruang	Positif	0	0	0	0
		Negatif	2	15	2	15

Confusion matrix pada Tabel 4.6 digunakan untuk mendapatkan evaluasi ketepatan klasifikasi senyawa obat kanker menggunakan LORENS yaitu akurasi dan AUC. Jumlah *True Positif* (TP), *False Positif* (FP), *False Negatif* (FN) dan *True Negatif* (TN) 5 subruang dengan *threshold* 0,5 berturut-turut yaitu 0 yang berarti tidak ada observasi kelas aktual positif yang tepat terklasifikasi positif, 2 yang berarti terdapat 2 observasi kelas aktual negatif terklasifikasi kedalam kelas positif, 2 yang menyatakan terdapat 2 pengamatan kelas aktual positif diklasifikasikan ke dalam kelas negatif dan 13 yang menyatakan

terdapat 13 kelas aktual negatif yang tepat terklasifikasi negatif. Partisi dengan subruang 8 *threshold* 0,5 tidak memiliki observasi kelas aktual positif yang tepat terklasifikasi positif, tidak terdapat observasi kelas aktual negatif terklasifikasi ke dalam kelas positif, terdapat 2 observasi kelas aktual positif terklasifikasi ke dalam kelas negatif, terdapat 15 observasi kelas aktual negatif tepat terklasifikasi negatif dan seterusnya hingga subruang 50 memiliki jumah observasi kelas aktual positif tepat diklasifikasikan positif sebesar 0, observasi kelas aktual negatif terklasifikasi kelas positif berjumlah 0, jumlah observasi kelas aktual positif diklasifikasikan kelas negatif sebesar 2 dan observasi kelas aktual negatif tepat terklasifikasi negatif berjumlah 15. Subruang dengan *threshold* optimum selanjutnya mengikuti proses tersebut. *Confusion matrix* yang ditunjukkan menunjukkan nilai *true positif* 0, yang berarti hasil yang didapatkan akan memperoleh nilai akurasi yang lebih cenderung tepat memprediksi kelas negatif yang mengakibatkan performansi kurang baik. Berikut merupakan nilai akurasi dan AUC klasifikasi LORENS *threshold* 0.5.

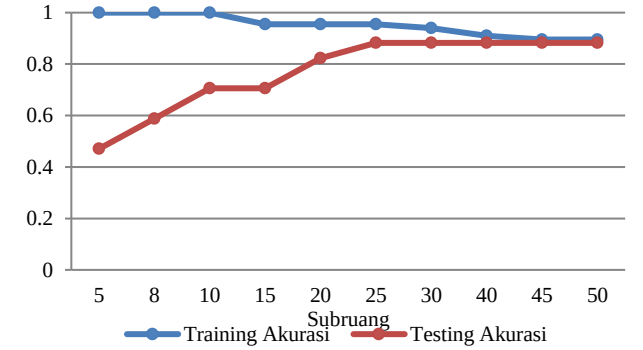


Gambar 4. 4 Akurasi LORENS *Threshold* 0,5

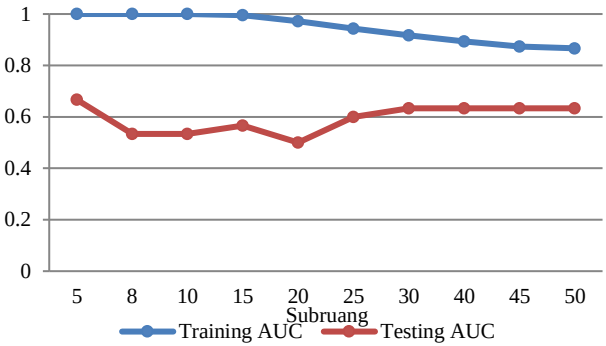


Gambar 4. 5 AUC LORENS *Threshold* 0,5

Berdasarkan Gambar 4.4 dan Gambar 4.5, dapat diketahui klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas dengan *threshold* 0,5 pada data *training* nilai akurasi terbesar adalah 1 atau 100% yaitu subruang 5, subruang 8 dan subruang 10. Begitu pula AUC yang memiliki nilai terbesar sebesar 1 pada ukuran subruang 5, subruang 8 dan subruang 10. Pada data *testing* nilai akurasi terbesar yaitu 88,24% yaitu pada kedelapan subruang yang digunakan kecuali subruang 5 dan subruang 10 dengan akurasi sebesar 70,59% dan 82,35%. Nilai AUC terbesar terletak pada 5 subruang sebesar 0,6667. Nilai AUC digunakan untuk mentukan jumlah subruang terbaik karena AUC tidak sensitif terhadap proporsi kelas.



Gambar 4. 6 Akurasi LORENS *Threshold* Optimum



Gambar 4. 7 AUC LORENS *Threshold* Optimum

Berdasarkan Gambar 4.6 dan Gambar 4.7 dapat diketahui klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas dengan metode LORENS *threshold* optimum menghasilkan nilai akurasi data *training* terbesar pada 5 subruang, 8 subruang dan 10 subruang dengan nilai 1 atau 100%. Sama seperti akurasi, nilai AUC pada data *training* memiliki nilai terbesar yaitu 1 pada ukuran 5 subruang, 8 subruang dan 10 subruang. Analisis LORENS *threshold* optimum pada data *testing* menunjukkan akurasi terbesar pada 25, 30, 45, dan 50 subruang yaitu sebesar 88,24%. Hasil perhitungan AUC data *testing* memiliki nilai terbesar adalah 0,6667 pada subruang berjumlah 5 subruang.

Berdasarkan Gambar 4.5 dan Gambar 4.7 pemilihan subruang terbaik dipilih dari nilai AUC terbesar pada data *testing* dengan *threshold* 0,5 dan *threshold* optimum. Jumlah subruang yang memiliki AUC terbesar adalah 5 subruang dengan *threshold* 0,5 dan *threshold* optimum sebesar 0,6667 sehingga jumlah subruang terbaik pada klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode LORENS adalah 5 subruang dengan *threshold* 0,5 maupun *threshold* optimum.

4.3.2 Analisis *Logistic Regression Ensemble* (LORENS) dengan *Synthetic Minority Oversampling Technique* (SMOTE)

Analisis LORENS dengan menggunakan SMOTE mengi-

kuti proses yang sama seperti Sub Bab 4.3.1 hingga mendapatkan prediksi kelas data *testing* yang merupakan hasil *majority voting* klasifikasi LORENS, namun terdapat perbedaan pada penggunaan data *training*. Pada sub bab sebelumnya data *training* yang digunakan *imbalance*, namun pada sub bab ini digunakan data *training* hasil SMOTE yaitu memiliki proporsi kelas yang *balance*. Pada Sub bab ini tidak dilakukan analisis LORENS dengan *threshold* optimum, karena proporsi kelas pada data *training* sudah seimbang atau *balance*. Berikut ini *confusion matrix* LORENS dengan SMOTE.

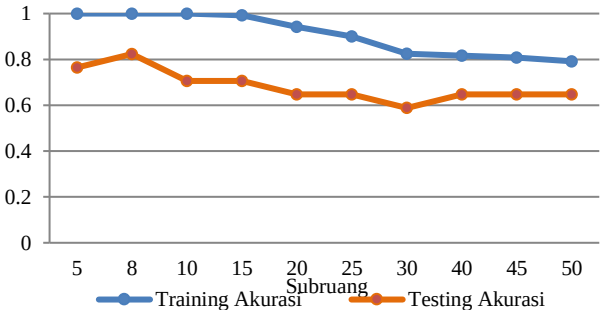
Tabel 4. 7 *Confusion Matrix* LORENS dengan SMOTE

			Kelas Aktual	
			Threshold 0,5	
			Positif	Negatif
Kelas Prediksi	5 Subruang	Positif	0	2
		Negatif	2	13
	8 Subruang	Positif	0	1
		Negatif	2	14
	10 Subruang	Positif	0	3
		Negatif	2	12
	15 Subruang	Positif	0	3
		Negatif	2	12
	20 Subruang	Positif	0	4
		Negatif	2	11
	25 Subruang	Positif	0	4
		Negatif	2	15
	30 Subruang	Positif	0	5
		Negatif	2	10
	40 Subruang	Positif	0	4
		Negatif	2	11

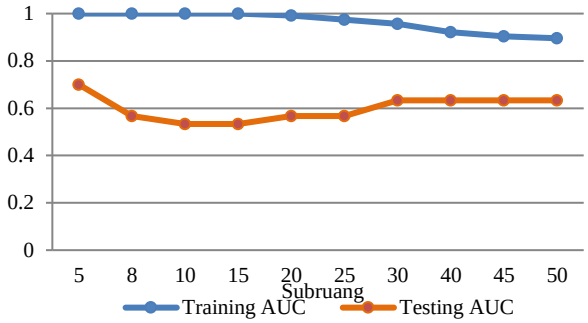
Tabel 4.7 *Confusion Matrix* LORENS dan SMOTE (Lanjutan)

			Kelas Aktual	
			Threshold 0,5	
			Positif	Positif
Kelas Prediksi	45 Subruang	Positif	0	4
		Negatif	2	11
	50 Subruang	Positif	0	4
		Negatif	2	11

Berdasarkan Tabel 4.7 dapat diketahui pada subruang berukuran 5 *threshold* 0,5 tidak terdapat observasi kelas aktual positif yang tepat diprediksi positif, terdapat 2 observasi kelas aktual negatif diprediksi positif, terdapat 2 observasi kelas aktual positif diprediksi negatif dan terdapat 13 observasi kelas aktual negatif yang tepat terprediksi kelas negatif dan seterusnya hingga subruang berukuran 50 *threshold* 0,5 yang tidak terdapat observasi kelas aktual positif tepat diprediksi positif, terdapat 4 observasi kelas aktual negatif diprediksi positif, terdapat 2 observasi kelas aktual positif diprediksi negatif dan 12 observasi kelas aktual negatif tepat diprediksi kelas negatif. *Confusion matrix* pada Tabel 4.7 menunjukkan tidak terdapatnya observasi kelas aktual positif tepat terprediksi positif yang berarti bahwa masih terdapat kekurangan dalam memprediksi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas yang dapat dianggap baik sebagai *radioprotector* dalam penyakit kanker.



Gambar 4. 8 Akurasi LORENS dengan SMOTE *Threshold* 0,5



Gambar 4. 9 AUC LORENS dengan SMOTE *Threshold* 0,5

Berdasarkan Gambar 4.8 dan Gambar 4.9 dapat diketahui klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode LORENS dengan SMOTE menunjukkan pada data *training* nilai akurasi dan AUC tertinggi yaitu subruang berukuran 5 subruang, 8 subruang dan 10 subruang sebesar 100%. Pada data *testing* nilai akurasi terbesar yaitu 82,35% yaitu subruang berukuran 8 subruang. Nilai AUC terbesar adalah 0,7 pada ukuran 5 subruang. Sehingga subruang optimal untuk digunakan adalah 5 subruang *threshold* 0,5

4.4 Klasifikasi Senyawa Obat Kanker Untuk Optimasi Proteksi Radiasi dan Toksisitas Menggunakan Metode *Ensemble Logistic Regression* (ELR)

Pada bagian ini akan dilakukan klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode *Ensemble Logistic Regression* (ELR). Analisis menggunakan data dengan variabel respon yang *imbalance* dan data variabel respon *balance* menggunakan hasil SMOTE dari Sub Bab 4.2, sehingga akan dilakukan analisis menggunakan ELR dan ELR dengan SMOTE.

4.4.1 Analisis *Ensemble Logistic Regression* (ELR)

Ensemble Logistic Regression (ELR) merupakan metode *ensemble* yang berbasis pada regresi logistik teregularisasi. Jenis regularisasi yang digunakan dalam penelitian ini yaitu regularisasi *l2*. Pada penelitian ini metode ELR akan dilakukan pengulangan sebanyak 10 kali. Dalam setiap pengulangan akan terdapat

beberapa iterasi hingga mendapatkan iterasi dengan model terbaik. Dalam menganalisis ELR langkah awal adalah menentukan parameter regulasi λ. Selanjutnya menghitung nilai inisialisasi vektor probabilitas awal dari setiap variabel dengan nilai 1-*p-value* dari *t-test ranking* yang bertujuan menentukan variabel prediktor yang akan digunakan dalam iterasi pertama untuk memodelkan data *training*. Jumlah variabel prediktor yang terpilih akan sama dengan jumlah observasi pada data *training* sehingga data yang digunakan tidak termasuk *high dimensional*. Nilai probabilitas tertinggi tidak dapat dipastikan terpilih sebagai variabel prediktor, namun kemungkinan besar dapat terpilih. Variabel dalam setiap iterasi metode ELR memiliki kemungkinan yang berbeda untuk terpilih sebagai variabel prediktor. Berikut ini diberikan Tabel 4.8 mengenai probabilitas awal dari masing-masing variabel prediktor.

Tabel 4. 8 Vektor Probabilitas Awal

No	Variabel	Probabilitas
1	pKa.max20.	0,9882
2	Br_Count	0,1185
3	C_Count	0,9640
4	Cl_Count	0,1185
5	F_Count	0,6534
6	H_Count	0,9077
7	I_Count	0,3750
⋮	⋮	⋮
217	Molecular_Volume	0,9527

Hal berikutnya menentukan nilai *Balance Classification Rate* (BCR) awal yang didapatkan dari rata-rata jumlah kelas positif. Nilai BCR_0 yang digunakan sebesar 0,107 karena data yang digunakan *imbalance*, lain halnya jika data yang digunakan *balance* maka nilai BCR_0 adalah 0,5. Analisis ELR untuk iterasi pertama sampai iterasi terakhir yaitu telah memperoleh nilai $\overline{BCR}$ yang konvergen yaitu ε kurang dari 10^{-5} dilakukan sebagai berikut. Membagi data menjadi 2 yaitu data *training* dan *testing* secara *stratified* dengan proporsi 80% data *training* dan 20% data

testing. Selanjutnya mendapatkan variabel prediktor yang digunakan dengan inisialisasi vektor probabilitas awal. Variabel prediktor yang terpilih pada iterasi pertama diberikan pada tabel dibawah ini.

Tabel 4. 9 Variabel Terpilih Berdasarkan Inisialisasi Vektor Probabilitas Awal

No	Variabel	Probability
1	pKa.max20.	0,9882
2	H_Count	0,9077
3	S_Count	0,9318
4	ALogP98	0,5658
5	ALogP_MR	0,9555
6	ES_Count_aasC	0,8266
7	ES_Count_ddsN	0,8185
8	ES_Count_dssC	0,9619
⋮	⋮	⋮
67	Molecular_3D_SAVol	0,9807

Berdasarkan Tabel 4.9 dapat diketahui variabel terpilih pada iterasi pertama klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan tokisitas dengan metode ELR yaitu pKa.max20. dengan probabilitas 0,9882, probabilitas H_Count sebesar 0,9077, probabilitas S_Count sebesar 0,9318 dan seterusnya sampai variabel terakhir yang terpilih yaitu Molecular_3D_SAVol sebesar 0,9807. Setelah mendapatkan variabel prediktor terpilih dilakukan pemodelan dengan variabel tersebut. Pemodelan dilakukan untuk setiap iterasi yang terjadi. Berikut ini diberikan model pada iterasi pertama dengan variabel terpilih pada Tabel 4.10.

Tabel 4. 10 Koefisien Regresi Iterasi Pertama

No	Parameter	Estimasi
1	β_0	-0,21411
2	β_1	-6,67E-07
3	β_6	-8,62E-08

Tabel 4.10 Koefisien Regresi Iterasi Pertama (Lanjutan)

No	Parameter	Estimasi
4	β_{10}	-6,67E-08
5	β_{11}	7,85E-08
6	β_{12}	-6,85E-07
7	β_{18}	2,87E-09
8	β_{19}	-1,64E-08
9	β_{23}	-4,16E-08
⋮	⋮	⋮
68	β_{216}	-5,47E-06

Tabel 4.10 diatas menunjukkan koefisien regresi variabel terpilih pada iterasi pertama metode ELR. Nilai β_0 yang dihasilkan yaitu -0,21411, nilai β_1 yaitu -6,67x10^{-7} dan seterusnya hingga β_{216} sebesar -5,47x10^{-6}. Setelah model terbentuk, data *testing* diaplikasikan terhadap model tersebut untuk mendapatkan prediksi data *testing* pada iterasi pertama.

Tabel 4. 11 Prediksi Data *Testing* Iterasi Pertama

No	Probability	Prediksi
1	0,0326	0
2	0,1732	0
3	0,0935	0
4	0,0855	0
5	0,1021	0
6	0,1020	0
7	0,0525	0
8	0,1024	0
9	0,0893	0
10	0,0857	0
11	0,0803	0
12	0,1910	0
13	0,1630	0
14	0,0269	0

Tabel 4.11 Prediksi Data *Testing* Iterasi Pertama (Lanjutan)

No	Probability	Prediksi
15	0,0430	0
16	0,0876	0
17	0,0252	0

Berdasarkan Tabel 4.11 diketahui bahwa tidak ada observasi data *testing* yang memiliki nilai probabilitas lebih dari nilai *threshold* sebesar 0,5. Hal ini menyebabkan nilai prediksi kelas data *testing* bernilai 0 atau negatif. Nilai prediksi ini akan digunakan untuk mendapatkan nilai *Balance Classification Rate* (BCR) melalui *confusion matrix*.

Tabel 4. 12 *Confusion Matrix* ELR Iterasi Pertama

		Kelas Aktual	
		Positif	Negatif
Kelas Prediksi	Positif	0	0
	Negatif	2	15

Berdasarkan Tabel 4.12 dapat dihitung nilai BCR iterasi pertama menggunakan Persamaan 2.24. Nilai BCR yang dihasilkan berdasarkan perhitungan sebesar 0,5. Hal berikutnya adalah menghitung nilai *quality* berdasarkan Persamaan 2.23. Nilai *quality* hasil perhitungan adalah 0,1439. Kedua nilai tersebut BCR dan *quality* digunakan untuk menghitung vektor probabilitas baru menggunakan Persamaan 2.22. Vektor probabilitas baru digunakan sebagai probabilitas baru dari variabel terpilih yang akan digunakan pada iterasi selanjutnya jika hasil perhitungan ε kurang dari 10^{-5}.

Tabel 4. 13 Vektor Probabilitas Baru Iterasi Pertama Metode ELR

No	Variabel	Probability Baru
1	pKa.max20.	0,9882
2	H_Count	0,9077
3	S_Count	0,9318
4	ALogP98	0,5658

Tabel 4.13 Vektor Probabilitas Baru Iterasi Pertama Metode ELR (Lanjutan)

No	Variabel	Probability Baru
5	ALogP_MR	0,9555
6	ES_Count_aasC	0,8266
7	ES_Count_ddsN	0,8185
8	ES_Count_dssC	0,9619
⋮	⋮	⋮
67	Molecular_3D_SAVol	0,9807

Berdasarakan Tabel 4.13 dapat diketahui bahwa variabel yang terpilih nilai vektor probabilitas telah diperbaharui. Variabel yang tidak terpilih pada iterasi pertama nilai vektor probabilitasnya tidak diperbaharui. Selanjutnya menghitung nilai $\overline{BCR}_1$ menggunakan Persamaan 2.25. Nilai $\overline{BCR}_1$ didapatkan sebesar 0,3036. $\overline{BCR}_0$ dan $\overline{BCR}_1$ kemudian dihitung selisihnya untuk mendapatkan nilai ε. Nilai ε yang didapatkan adalah 0,1964. Hasil perhitungan dari nilai ε menunjukkan hasil yang lebih besar dari 10^{-5}. Oleh karena itu, iterasi kedua akan dilakukan dengan proses yang sama seperti iterasi pertama hingga mendapatkan nilai ε kurang dari 10^{-5}.

Tabel 4. 14 Variabel Terpilih Berdasarkan Inisialisasi Vektor Probabilitas Baru Iterasi Pertama

No	Variabel	Probability
1	C_Count	0,9640
2	F_Count	0,6534
3	ALogP_MR	0,9555
4	ES_Count_aaN	0,6468
5	ES_Count_aaO	0,6794
6	ES_Count_aasC	0,8266
7	ES_Count_dssC	0,9619
8	ES_Count_sF	0,6534
⋮	⋮	⋮
67	Shadow_YZfrac	0,6729

Tabel 4.14 menunjukkan variabel yang terpilih pada iterasi kedua. Variabel prediktor yang terpilih yaitu C_Count dengan nilai probability 0,9640, F_Count sebesar 0,6534, AlogP_MR dengan nilai 0,9555 hingga variabel prediktor terakhir yang terpilih yaitu Shadow_YZfrac dengan probability 0,6729. Kemudian memodelkan variabel yang terpilih dengan koefisien regresi yang didapatkan sebagai berikut.

Tabel 4. 15 Koefisien Regresi Iterasi Kedua

No	Parameter	Estimasi
1	β_0	-1,18472
2	β_3	-3,47E-07
3	β_5	3,18E-07
4	β_{12}	-7,62E-06
5	β_{15}	7,37E-07
6	β_{17}	-1,46E-08
7	β_{18}	2,96E-08
8	β_{23}	-3,21E-07
9	β_{27}	3,81E-07
⋮	⋮	⋮
68	β_{212}	7,08E-08

Berdasarkan Tabel 4.15 didapatkan nilai koefisien regresi pada iterasi kedua dengan nilai β_0 sebesar -1,18472, β_3 sebesar -3,47x10^{-07} dan seterusnya hingga nilai β_{212} sebesar 7,08x10^{-08}. Setelah didapatkan model regresi logistik terregularisasi, model diterapkan dengan data *testing* untuk medapatkan prediksi data *testing*. Berikut ini prediksi data *testing* pada iterasi kedua.

Tabel 4. 16 Prediksi Data *Testing* Iterasi Kedua

No	Probability	Prediksi
1	0.0310	0
2	0.1823	0
3	0.1086	0
4	0.0979	0
5	0.0975	0

Tabel 4.16 Prediksi Data *Testing* Iterasi Kedua (Lanjutan)

No	Probability	Prediksi
6	0.1166	0
7	0.0551	0
8	0.1026	0
9	0.0929	0
10	0.0832	0
11	0.0996	0
12	0.1754	0
13	0.1672	0
14	0.0210	0
15	0.0355	0
16	0.1171	0
17	0.0184	0

Berdasarkan Tabel 4.16 dapat diketahui bahwa nilai probability observasi pertama pada data *testing* sebesar 0,0310 yang lebih kecil dari nilai *threshold* 0,5sehingga prediksi kelas data testing observasi pertama bernilai negatif atau 0, observasi kedua memiliki nilai probability sebesar 0,1823 yang kurang dari nilai *threshold* sehingga prediksi bernilai negatif atau 0 hingga observasi terakhir yang memiliki nilai probability 0,0184 kurang dari *threshold* maka prediksi menghasilkan nilai negatif atau 0. Setelah didapatkan prediksi data *testing*, akan diberikan *confusion matrix* untuk menghitung nilai BCR pada iterasi kedua.

Tabel 4. 17 *Confusion Matrix* ELR Iterasi Kedua

		Kelas Aktual	
		Positif	Negatif
Kelas Prediksi	Positif	0	0
	Negatif	2	15

Berdasarkan Tabel 4.17 dengan menggunakan Persamaan 2.24 dapat dihitung nilai BCR pada iterasi kedua. Nilai BCR yang didapatkan sebesar 0,5. Kemudian menghitung nilai *quality* berdasarkan Persamaan 2.23 yaitu sebesar 0,0779. Setelah

didapatkan nilai BCR dan *quality*, selanjutkan menghitung vektor probabilitas baru berdasarkan Persamaan 2.22 dengan hasil sebagai berikut.

Tabel 4. 18 Vektor Probabilitas Baru Iterasi Kedua Metode ELR

No	Variabel	Probability
1	C_Count	0.9643
2	F_Count	0.6536
3	ALogP_MR	0.9558
4	ES_Count_aaN	0.6470
5	ES_Count_aaO	0.6796
6	ES_Count_aasC	0.8268
7	ES_Count_dssC	0.9622
8	ES_Count_sF	0.6536
⋮	⋮	⋮
67	Shadow_YZfrac	0,6731

Berdasarkan Tabel 4.18 vektor probabilitas telah diperbaharui untuk variabel terpilih pada iterasi kedua. Variabel yang tidak terpilih pada iterasi kedua, nilai probabilitasnya tetap atau tidak berubah. Selanjutnya menghitung nilai $\overline{BCR}_2$ menggunakan Persamaan 2.25 didapatkan hasil sebesar 0,3690. Kemudian menghitung selisih nilai $\overline{BCR}_1$ dan $\overline{BCR}_2$ untuk mendapatkan nilai ε. Hasil nilai ε yang didapatkan sebesar 0,0655. Nilai ε hasil perhitungan menunjukkan angka yang lebih besar dari 10^{-5}. Karena nilai ε masih lebih besar dari 10^{-05}, iterasi selanjutnya akan dilakukan dengan proses yang sama seperti iterasi pertama dan kedua hingga mendapatkan nilai ε kurang dari 10^{-5}.

Model terbaik didapatkan dari iterasi terakhir yang menghasilkan nilai ε kurang dari $10^{-}5$. Berikut ini hasil model terbaik yang didapatkan dari iterasi terakhir.

Tabel 4. 19 Koefisien Regresi Metode ELR Model Terbaik

No	Parameter	Estimasi
1	β_0	-1,2353

Tabel 4.19 Koefisien Model Regresi ELR Model Terbaik (Lanjutan)

No	Parameter	Estimasi
2	β_1	-1,12E-06
3	β_3	-5,52E-08
4	β_7	-1,25E-08
5	β_8	9,30E-08
6	β_{11}	1,33E-07
7	β_{12}	-1,42E-06
8	β_{21}	-3,58E-07
9	β_{23}	-5,11E-08
⋮	⋮	⋮
67	β_{215}	-1,08E-05

Berdasarkan Tabel 4.19 dapat diketahui model terbaik yang dihasilkan iterasi terakhir. Nilai β_0 sebesar -1,2353, nilai β_1 sebesar -1,12x10^{-6}, nilai β_3 sebesar -5,52x10^{-8} hingga nilai β_{215} sebesar -1,08x10^{-5}. Model terbaik yang didapatkan kemudian digunakan untuk memprediksi kelas pada data *testing*.

Tabel 4. 20 Prediksi Data *Testing* Metode ELR Model Terbaik

No	Probability	Prediksi
1	0,0284	0
2	0,1702	0
3	0,1096	0
4	0,0992	0
5	0,1094	0
6	0,1198	0
7	0,0617	0
8	0,1095	0
9	0,0994	0
10	0,0892	0
11	0,0793	0
12	0,1783	0
13	0,1701	0

Tabel 4.20 Prediksi Data *Testing* Metode ELR Model Terbaik (Lanjutan)

No	Probability	Prediksi
14	0,0276	0
15	0,0451	0
16	0,1200	0
17	0,0159	0

Tabel 4.20 menunjukkan hasil prediksi data *testing* menggunakan model terbaik. Jika nilai probabilitas lebih besar dari *threshold* 0,5 maka prediksi kelas data *testing* adalah positif atau satu, namun apabila nilai probabilitas lebih kecil dari 0,5 kelas prediksi data *testing* bernilai 0 atau negatif. Dapat dilihat bahwa nilai probabilitas observasi pertama yaitu 0,0284 lebih kecil dari 0,5 sehingga memiliki nilai prediksi kelas negatif atau 0, probabilitas observasi kedua sebesar 0,1702 lebih kecil dari 0,5 sehingga nilai prediksi bernilai 0, dan seterusnya hingga observasi terakhir nilai probabilitas sebesar 0,0159 lebih kecil dari 0,5 sehingga nilai prediksi kelas negatif.

Tabel 4. 21 *Confusion Matrix* Metode ELR Model Terbaik

		Kelas Aktual	
		Positif	Negatif
Kelas Prediksi	Positif	0	0
	Negatif	2	15

Tabel 4.21 merupakan *confusion matrix* dari hasil klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode ELR pada pengulangan pertama dengan model terbaik. Jumlah observasi kelas aktual positif tepat diprediksi positif dan observasi kelas aktual negatif yang diprediksi positif berjumlah 0, jumlah observasi kelas aktual positif yang diprediksi negatif adalah 2 observasi dan jumlah observasi kelas aktual negatif tepat diprediksi negatif adalah 15 observasi. Nilai prediksi kelas aktual positif yang tepat terklasifikasi positif berjumlah 0 yang menyatakan bahwa hasil klasifikasi yang diperoleh cenderung memiliki performa yang kurang baik karena kelas negatif memiliki kontribusi lebih

banyak dibandingkan kelas positif dimana jumlah kelas aktual negatif yang tepat terprediksi negatif sebanyak 15 observasi. Pada pengulangan kedua hingga kesepuluh, proses mendapatkan model terbaik hingga mendapatkan prediksi kelas data *testing* sama seperti pengulangan pertama.

Tabel 4. 22 Hasil Klasifikasi Metode ELR

Pengulangan	Training		Testing	
	Akurasi	AUC	Akurasi	AUC
1	0,8955	0,7667	0,8824	0,6000
2	0,8955	0,7476	0,8824	0,5333
3	0,8955	0,7476	0,8824	0,5333
4	0,8955	0,7452	0,8824	0,5667
5	0,8955	0,7619	0,8824	0,6000
6	0,8955	0,5000	0,8824	0,5000
7	0,8955	0,5000	0,8824	0,5000
8	0,8955	0,7643	0,8824	0,6000
9	0,8955	0,7667	0,8824	0,6000
10	0,8955	0,7738	0,8824	0,6000
Rata-rata	0,8955	0,7074	0,8824	0,5633

Bersarkan Tabel 4.22 dapat dilihat bahwa hasil klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode ELR memiliki nilai total akurasi pada data *training* dan *testing* dirata-rata untuk kesepuluh pengulangan yang dilakukan. Hasil total memiliki nilai akurasi data *training* sebesar 89,95% dan akurasi data *testing* sebesar 88,24% serta nilai AUC data *training* sebesar 0,7074 dan nilai AUC data *testing* sebesar 0,5633.

4.4.2 Analisis *Ensemble Logistic Regression* (ELR) dengan *Synthetic Minority Oversampling Technique* (SMOTE)

Analisis klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan ELR dengan *Synthetic Minority Overssampling Method* (SMOTE) menjalani proses yang sama seperti Sub Bab 4.4.1 dengan 10 pengulangan.

Berikut ini model terbaik dari pengulangan pertama yang didapatkan menggunakan Metode ELR dengan SMOTE.

Tabel 4. 23 Koefisien Regresi Metode ELR dengan SMOTE Model Terbaik

No	Parameter	Estimasi
1	β_0	1,689794
2	β_1	-5,01E-05
3	β_3	5,88E-06
4	β_5	2,35E-06
5	β_6	1,13E-05
6	β_8	7,16E-06
7	β_9	-9,36E-06
8	β_{12}	3,04E-05
9	β_{15}	4,73E-06
⋮	⋮	⋮
67	β_{216}	3,63E-05

Berdasarkan Tabel 4.23 dapat dilihat koefisien regresi metode ELR dengan SMOTE pada model terbaik. Nilai β_0 adalah 2,5154, nilai β_1 adalah -5,01x10^{-5}, nilai β_3 adalah 5,88x10^{-6} dan seterusnya hingga nilai β_{216} adalah 3,63x10^{-5}.

Tabel 4. 24 Prediksi Data *Testing* Metode ELR dengan SMOTE Model Terbaik

No	Probability	Prediksi
1	0,0243	0
2	0,7460	1
3	0,3875	0
4	0,3347	0
5	0,3367	0
6	0,4218	0
7	0,0836	0
8	0,3570	0
9	0,3212	0
10	0,2659	0

Tabel 4.24 Prediksi Data *Testing* Metode ELR dengan SMOTE Model Terbaik (Lanjutan)

No	Probability	Prediksi
11	0,4693	0
12	0,6984	1
13	0,7120	1
14	0,0072	0
15	0,0339	0
16	0,4230	0
17	0,0059	0

Tabel 4.24 menunjukkan prediksi data *testing* metode ELR dengan SMOTE pada model terbaik pengulangan pertama. Observasi pertama memiliki probabilitas sebesar 0,0243 sehingga nilai prediksi kelas observasi pertama yaitu negatif atau 0, observasi kedua memiliki probabilitas sebesar 0,7460 sehingga prediksi kelas bernilai positif atau 1, begitu seterusnya hingga observasi terakhir dengan nilai probabilitas sebesar 0,0059 sehingga nilai prediksi kelas bernilai negatif atau 0.

Tabel 4. 25 *Confusion Matrix* Metode ELR Dengan SMOTE Model Terbaik

		Kelas Aktual	
		Positif	Negatif
Kelas Prediksi	Positif	0	3
	Negatif	2	12

Berdasarkan Tabel 4.25 dapat dilihat hasil *confusion matrix* klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode ELR dengan SMOTE model terbaik pada pengulangan pertama bahwa tidak ada observasi kelas aktual positif yang tepat teprediksi positif dimana hal ini dapat mempengaruhi ketepatan klasifikasi karena kelas positif akan memiliki pengaruh yang sangat rendah sehingga menyebabkan performansi menjadi kurang baik.Terdapat 3 observasi kelas aktual negatif yang terprediksi positif, ada 2

observasi kelas aktual positif yang terklasifikasi negatif dan ada 12 kelas aktual negatif tepat terprediksi kelas negatif.

Tabel 4. 26 Hasil Klasifikasi Metode ELR Dengan SMOTE

Pengulangan	Training		Testing	
	Akurasi	AUC	Akurasi	AUC
1	0,7583	0,7839	0,7059	0,6333
2	0,7583	0,7703	0,7059	0,6000
3	0,7333	0,7800	0,7059	0,6333
4	0,7667	0,7831	0,7059	0,6667
5	0,7583	0,7850	0,7059	0,6667
6	0,7667	0,7850	0,7059	0,6667
7	0,7333	0,7606	0,7059	0,5333
8	0,7333	0,7783	0,7059	0,6000
9	0,7500	0,7611	0,7059	0,6000
10	0,7750	0,7883	0,7059	0,6667
Rata-rata	0,7533	0,7776	0,7059	0,6267

Tabel 4.26 menunjukkan hasil klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode ELR dengan SMOTE untuk kesepuluh pengulangan. Hasil yang digunakan adalah rata-rata dari evaluasi ketepatan model dari pengulangan pertama hingga sepuluh. Nilai akurasi data *training* dan data *testing* sebesar 75,33% dan 70,59%. Nilai AUC data *training* dan data *testing* sebesar 0,7776 dan 0,6267.

4.5 Pemilihan Metode Terbaik

Pemilihan metode terbaik dalam klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode *Logistic Regression Ensemble* (LORENS) dan *Ensemble Logistic Regression* (ELR) dengan *Synthetic Minority Oversampling Technique* (SMOTE) dan tanpa SMOTE dilakukan dengan membandingkan metode tersebut. Metode terbaik ditentukan dari nilai AUC terbesar yang diperoleh. Nilai AUC baik digunakan untuk data yang *imbalance* karena AUC

tidak sensitif terhadap proporsi kelas variabel respon. Metode yang dibandingkan yaitu metode LORENS, metode LORENS dengan SMOTE, metode ELR dan metode ELR dengan SMOTE.

Tabel 4. 27 Pemilihan Metode Terbaik

		Training		Testing	
		Akurasi	AUC	Akurasi	AUC
LORENS	5 Subruang *Threshold* 0,5	1.0000	1.0000	0.7647	0.6667
	5 Subruang *Threshold* Optimum	1.0000	1.0000	0.4706	0.6667
	SMOTE 5 Subruang *Threshold* 0,5	1.0000	1.0000	0.7647	0.7000
ELR	ELR	0,8955	0,7074	0,8824	0,5633
	ELR-SMOTE	0,7533	0,7776	0,7059	0,6267

Berdasarkan Tabel 4.27 dapat diketahui metode LORENS dengan SMOTE 5 subruang *threshold* 0,5 memiliki nilai AUC terbaik dibandingkan dengan metode lainnya. Klasifikasi senyawa obat kanker untuk oprimasi proteksi radiasi dan toksisitas menggunakan LORENS 5 subruang *threshold* 0,5 memiliki nilai AUC tertinggi bernilai 0,7.

(Halaman ini sengaja dikosongkan)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut.

1. Proporsi kelas variabel respon baru menunjukkan *imbalance* sebesar 9 kelas positif dengan 75 kelas negatif dimana kelas positif yaitu memiliki tingkat kematian sel kanker tinggi pada radiasi proteksi dan tingkat kematian sel normal rendah pada toksisitas dan kelas negatif selainnya. Terdapat perbedaan nilai rata-rata variabel prediktor kelas positif atau kelas 1 dengan kelas negatif atau kelas 0.
2. *Synthetic Minority Oversampling Thecnique* (SMOTE) mampu mengatasi data *imbalance*. SMOTE hanya digunakan pada data *training*. Proporsi kelas data *training* variabel respon memiliki proporsi 7:60 dengan 7 positif dan 60 negatif. Setelah diaplikasikan metode SMOTE proporsi kelas data *training* variabel respon menjadi *balance* atau seimbang yaitu 60 positif dan 60 negatif.
3. Klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas menggunakan metode LORENS menghasilkan subruang terbaik berjumlah 5 menggunakan *threshold* 0,5 dan optimum dengan nilai AUC sebagai evaluasi ketepatan klasifikasi sebesar 0,6667. Menggunakan metode LORENS dengan SMOTE mendapatkan subruang terbaik berjumlah 5 dengan *threshold* 0,5 menghasilkan nilai AUC sebesar 0,7.
4. Metode *Ensemble Logistic Regression* (ELR) dalam klasifikasi senyawa obat kanker untuk optimasi proteksi radiasi dan toksisitas memiliki nilai evaluasi ketepatan klasifikasi AUC sebesar 0,5633. Penggunaan SMOTE pada metode ELR memiliki nilai AUC yang berbeda dengan ELR tanpa SMOTE yaitu sebesar 0,6267.
5. Metode klasifikasi terbaik yang menghasilkan nilai AUC tertinggi antara LORENS dan ELR dengan SMOTE dan tanpa SMOTE yaitu metode LORENS dengan SMOTE

menggunakan 5 subruang *threshold* 0,5 dengan nilai AUC sebesar 0,7.

5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya antara lain melakukan *split data* dapat digunakan metode lain selain *holdout* sehingga data yang digunakan memiliki kesempatan sebagai data *training* dan data *testing* dengan harapan mampu meningkatkan hasil evaluasi ketepatan model. Mengkaji lebih lanjut terhadap hasil klasifikasi yang didapatkan dengan data *imbalance*. Meskipun data *imbalance* sudah ditangani sehingga *balance* namun pada penelitian ini masih menunjukkan performansi yang perlu diperbaiki, dimana kemampuan prediksi untuk kelas positif masih rendah. Hal ini ditunjukkan dari nilai akurasi yang secara umum lebih didominasi oleh kemampuan prediksi untuk kelas negatif.

DAFTAR PUSTAKA

Ariyasu, S., Sawa, A., Morita, A., Hanaya, K., Hoshi, M., Takahashi, I., & Aoki, S. (2014). Design and Synthesis of 8-Hydroxyquinoline-Based Radioprotective Agents. *Bioorganic and Medicinal Chemistry, 22*(15), 3891-3905.

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

Fawcett, T. (2006). An introduction to ROC Analysis. *Pattern Recognition Letters, 27*, 861-874.

Fithrony, M. (2012). *Pengaruh Radioterapi Area Kepala Dan Leher Terhadap Curah Saliva.* Semarang: Program Pendidikan Sarjana Kedokteran Fakultas Kedokteran Universitas Diponegoro.

Gilbert, S. (1991). *Developmental Biology.* Massachusetts: Sinauer Associaters.

Han, J., & Kamber, M. (2006). *Data Mining Concepts And Techniques.* San Francisco: Morgan Kaufmann Publisher.

He, H., Member, IEEE, & Garcia, E. A. (2009). Learning From Imbalance Data. *IEEE Transactions On Knowledge And Data Engineering, 21*(9), 1263-1284.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression* (2nd ed.). New York: John Willey And Sons. doi:10.1002/0471722146

Kuswanto, H., & Werdhana, R. W. (2018). Logistic Regression Ensemble to Classify Alzheimer Gene Expression. *Internetworking Indonesia Journal, 10*(1), 36-41.

Kuswanto, H., Asfirani, A., Sarumaha, Y., & Ohwada, H. (2015). Logistic Regression Ensemble for Predicting Customer

Defection with Very Large Sample Size. *Procedia Computer Science, 72*, 86-93.

Lee, K., Ahn, H., Moon, H., Kodell, R., & Chen, J. (2013). Multinomial Logistic Regression Ensembles. *Journal of Biopharmaceutical Statistics, 23*(3), 681-694.

Lim, N., Ahn, H., Moon, H., & Chen, J. J. (2010). Classification of High-Dimensional Data With Ensemble of Logistic Regression Models. *Journal of Biopharmaceutical Statistics, 20*, 160-171.

Mangan, Y. (2009). *Solusi Sehat Mencegah Dan Mengatasi Kanker*. Jakarta: Agromedia Pustaka.

Matsumoto, A., Aoki, S., & Ohwada, H. (2016). Comparison of Random Forest and SVM for Raw Data in Drug Discovery: Prediction of Radiation Protection and Toxicity Case Study. *International Journal of Machine Learning and Computing, 6*(2), 145-148.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5 ed.). New Jersey: John Wiley and Sons.

Morita, A., Ariyasu, S., Wang, B., Asamaru, T., Onoda, T., Sawa, A., . . . Aoki, S. (2014). A Novel Inhibitor of p53-Dependent Apoptosis, Prevents Apoptotic Mitochondrial Dysfunction in A Transcription-Dependent Manner and Protects Mice From A Lethal Dose of Ionizing Radiation. *Biochemical and Biophysical Research Communications, 450*, 1498-1504.

Mubarok, R. (2018). *Klasifikasi Senyawa Obat Kanker Untuk Optimasi Proteksi Radiasi Menggunakan Pendekatan Machine Learning.* Surabaya: Program Studi Sarjana Departemen Statistika Fakultas Matematika, Komputasi dan Sains Data ITS.

Ng, A. Y. (2004). Feature Selection, L1 vs L2 Regularization and Rotational Invariance. *Proceedings of The Twenty-First International Conference on Machine Learning, 78*, 1-4..

Okun, O. (2011). *Feature Selection and Ensemble Methods for Bioinformatics: Algoritm Classification and Implementation.* United States of America: IGI Global.

Organization, W. H. (2009). *Cancer*. Dipetik Februari 12, 2019, dari World Health Organization: http://www.who.int/news-room/fact-sheets/detail/cancer

Pangastuti, S. S. (2018). *Perbandingan Metode Ensemble Random Forest dengan SMOTE-Boosting dan SMOTE-Bagging Pada Replikasi Data Mining Untuk Kelas Imbalance*. Surabaya: Program Magister Departemen Statistika Fakultas Matematika, Komputasi dan Sains Data ITS.

Purnami, S. W., Khasanah, P. M., Sumartini, S. H., Chosuvivatwong, V., & Sriplung, H. (2016). Cervical Cancer Survival Prediction Using Hybrid of SMOTE, CART, and Smooth Support Vector Machine. *AIP Conference Proceedings, 1723*, 1-8. doi:10.1063/1.4945075

RI, D. (2017). *Kementrian Kesehatan Ajak Masyarakat Cegah Dan Kendalikan Kanker*. Dipetik November 24, 2018, dari Departemen Kesehatan RI: http://www.depkes.go.id/article/view/17020200002/kementrian-kesehatan-ajak-masyarakat-cegah-dan-kendalikan-kanker.html

Siegel, R. L., Miller, K. D., & Jemal, A. (2019). CA Cancer J Clin. *Cancer Statistics, 69*, 7-34.

Sukmaputri, E. (2018). *Klasifikasi Senyawa Obat Kanker Untuk Optimasi Toksisitas Menggunakan Logistic Regression Ensembles (LORENS) Dan Ensemble of Support Vector*

Machine Dengan Feature Selection Pada High Dimensional Data. Surabaya: Program Studi Sarjana Departemen Statistika Fakultas Matematika, Komputasi dan Sains Data ITS.

WCRF, & AICR. (1997). *Food, Nutrition and the Prevention of Cancer.* London: a Global Perspective.

Widhianingsih, T. D. (2018). *Klasifikasi Data Berdimensi TInggi Dengan Metode Ensemble Berbasis Regresi Logistik Dalam Permasalahan Drug Discovery.* Surabaya: Program Magister Departemen Statistika Fakultas Matematika, Komputasi dan Sains Data ITS.

William, & Gilbert. (1991). Programmed Cell Death: Apoptosis and Oncogenesis. *65*, 1097-1098.

Yan, X., & Su, X. G. (2009). *Linear Regression Analysis Theory and Computing.* Singapore: World Scientific Pub Co Inc.

Zakharov, R., & Dupont, P. (2011). Ensemble Logistic Regression for Feature Selection. *Pattern Recognition in Bioinformatics, 7036*, 133-144.

Zhang, D., Liu, W., Gong, X., & Jin, H. (2011). A Novel Improved SMOTE Resampling Algorithm Based On Fractal. *Computational Information Systems, 7*, 2204-2211.

Zhou, Z.-H. (2012). *Ensemble Methods Foundations and Algorithms.* New York, United States: CRC Press.

LAMPIRAN

Lampiran 1. Data Senyawa Obat Kanker

Y	pKa(max20)	Br_Count	C_Count	H_Count	⋯	Molecular_ Volume
0	20	0	19	29	⋯	351.91
0	11.4	0	11	13	⋯	224.66
0	7.9	0	10	10	⋯	170.81
1	8.3	0	11	12	⋯	188.3
0	20	0	11	11	⋯	140.62
0	20	0	12	14	⋯	202.36
0	10.1	0	13	17	⋯	261.02
0	20	0	13	17	⋯	288.11
0	11.5	0	12	15	⋯	244.9
0	20	0	10	8	⋯	139.6
0	20	0	11	10	⋯	154.69
0	20	0	13	16	⋯	222.94
0	20	0	14	18	⋯	234.95
1	6.4	0	12	13	⋯	228.43
0	20	0	16	23	⋯	311.78
0	20	0	20	31	⋯	369.41
1	9.8	0	9	7	⋯	108.73
0	10.1	0	10	9	⋯	130.68
⋮	⋮	⋮	⋮		⋱	⋮
0	9.8	0	20	29	⋯	363.23

Lampiran 2. Variabel Prediktor LORENS 5 Subruang *Threshold* 0,5

Variabel	*Ensemble*										
	1	2	3	4	5	6	7	8	9	10	11
pKa.max20.	4	2	1	1	2	3	3	4	5	1	1
Br_Count	1	3	1	2	2	4	3	2	4	4	1
C_Count	1	3	2	3	5	3	2	2	1	3	1
Cl_Count	3	3	5	3	4	5	3	4	5	5	3
F_Count	3	1	3	2	3	4	5	5	2	3	3
H_Count	1	1	3	5	5	5	3	2	3	4	3
I_Count	3	3	2	1	4	3	1	5	1	4	2
N_Count	1	1	3	3	3	1	1	3	1	1	2
O_Count	2	4	5	4	4	3	2	1	4	2	1
S_Count	2	4	2	4	5	4	3	5	2	5	5
ALogP98	2	4	2	3	2	4	5	3	4	3	4
ALogP_MR	5	2	2	1	3	4	1	1	4	5	4
ES_Count_aaaC	2	4	5	1	3	5	1	1	1	1	2
ES_Count_aaCH	3	4	1	4	4	4	2	5	4	3	5
ES_Count_aaN	3	3	2	3	3	5	2	3	4	3	1
ES_Count_aaNH	1	4	3	1	5	4	2	4	5	2	4
ES_Count_aaO	1	2	4	2	3	5	5	1	1	3	4
ES_Count_aasC	3	1	4	1	2	5	4	5	2	1	3
ES_Count_ddsN	4	3	5	5	4	3	4	5	1	2	3
ES_Count_ddssS	1	5	4	3	4	1	5	3	2	4	5
ES_Count_dO	2	2	5	1	5	4	3	1	5	2	4
ES_Count_dsN	5	5	5	1	3	2	3	5	3	4	4
ES_Count_dssC	1	4	2	2	3	4	2	2	1	2	1
ES_Count_sBr	2	1	4	4	1	5	5	4	3	4	4
ES_Count_sCH3	2	3	3	4	4	5	4	1	3	4	5
⋮	⋮			⋮				⋮			⋮
Molecular_3D _PolarSASA	2	5	2	3	2	2	4	3	3	2	1

Lampiran 3. Variabel Prediktor LORENS 8 Subruang *Threshold* 0,5

Variabel	*Ensemble*										
	1	2	3	4	5	6	7	8	9	10	11
pKa.max20.	6	3	1	2	4	4	5	6	8	1	1
Br_Count	1	4	1	2	3	7	4	3	6	6	1
C_Count	1	5	4	5	8	5	2	3	1	5	1
Cl_Count	5	5	8	5	7	7	4	6	8	7	5
F_Count	4	1	5	4	5	6	7	7	3	5	4
H_Count	1	1	4	8	8	7	5	2	4	5	4
I_Count	5	5	3	2	6	4	2	8	2	6	3
N_Count	2	1	5	5	5	2	2	4	1	1	2
O_Count	3	7	7	7	7	4	2	1	7	3	1
S_Count	3	6	3	6	7	6	4	8	3	8	7
ALogP98	3	7	2	4	3	7	7	4	5	4	7
ALogP_MR	8	3	4	1	5	6	2	1	6	8	7
ES_Count_aaaC	2	5	8	2	5	7	2	1	2	1	3
ES_Count_aaCH	4	7	1	6	7	7	3	8	5	5	8
ES_Count_aaN	4	4	3	4	5	8	2	4	6	4	2
ES_Count_aaNH	1	5	4	1	8	6	3	6	7	2	6
ES_Count_aaO	2	3	6	2	4	8	8	1	2	5	6
ES_Count_aasC	5	2	6	1	3	8	5	8	4	2	5
ES_Count_ddsN	6	4	8	8	6	4	6	7	1	3	5
ES_Count_ddssS	1	8	7	5	6	2	8	4	2	7	8
ES_Count_dO	3	2	7	1	8	6	4	1	8	2	6
ES_Count_dsN	8	7	8	2	5	3	4	7	4	6	6
ES_Count_dssC	2	6	3	2	4	6	3	3	1	3	2
ES_Count_sBr	3	2	6	7	1	7	7	5	5	7	6
ES_Count_sCH3	4	4	4	7	6	8	5	1	4	5	7
⋮	⋮			⋮				⋮			⋮
Molecular_Volume	4	7	3	4	3	3	6	4	5	2	1

Lampiran 4. Variabel Prediktor LORENS 10 Subruang *Threshold* 0,5

Variabel	*Ensemble*										
	1	2	3	4	5	6	7	8	9	10	11
pKa.max20.	7	4	1	2	4	5	6	8	10	1	2
Br_Count	1	5	2	3	3	8	5	3	8	8	1
C_Count	1	6	4	6	9	6	3	3	1	6	1
Cl_Count	6	6	9	6	8	9	5	7	10	9	6
F_Count	5	1	6	4	6	7	9	9	4	6	5
H_Count	1	1	5	10	10	9	6	3	5	7	5
I_Count	6	6	3	2	8	5	2	10	2	7	3
N_Count	2	1	6	6	6	2	2	5	1	1	3
O_Count	4	8	9	8	8	5	3	1	8	3	1
S_Count	4	8	4	7	9	8	5	10	3	10	9
ALogP98	4	8	3	5	4	8	9	5	7	5	8
ALogP_MR	10	4	4	1	6	7	2	1	7	10	8
ES_Count_aaaC	3	7	10	2	6	9	2	1	2	1	3
ES_Count_aaCH	5	8	1	7	8	8	3	10	7	6	10
ES_Count_aaN	5	5	4	5	6	9	3	5	7	5	2
ES_Count_aaNH	1	7	5	1	10	7	4	7	9	3	8
ES_Count_aaO	2	3	7	3	5	10	10	2	2	6	7
ES_Count_aasC	6	2	7	1	4	9	7	10	4	2	6
ES_Count_ddsN	8	5	10	10	8	5	7	9	1	3	6
ES_Count_ddssS	2	10	8	6	7	2	9	5	3	8	10
ES_Count_dO	4	3	9	1	9	7	5	1	10	3	7
ES_Count_dsN	10	9	9	2	6	4	5	9	5	8	8
ES_Count_dssC	2	8	3	3	5	7	4	4	2	4	2
ES_Count_sBr	3	2	8	8	2	9	9	7	6	8	7
ES_Count_sCH3	4	5	5	8	8	10	7	1	5	7	9
⋮	⋮			⋮				⋮			⋮
Molecular_Volume	4	9	3	5	4	3	8	5	6	3	1

Lampiran 5. Variabel Prediktor LORENS 15 Subruang *Threshold* 0,5

Variabel	*Ensemble*										
	1	2	3	4	5	6	7	8	9	10	11
pKa.max20.	11	6	1	3	6	7	9	11	14	1	2
Br_Count	2	7	2	4	5	12	7	5	11	11	2
C_Count	1	9	6	9	14	9	4	5	1	9	1
Cl_Count	9	9	14	9	12	14	8	10	15	13	8
F_Count	8	1	8	6	9	10	13	13	6	9	7
H_Count	1	1	8	15	15	13	8	4	7	10	8
I_Count	9	9	5	3	11	7	3	15	3	10	5
N_Count	3	1	9	8	9	3	3	7	1	2	4
O_Count	6	12	13	12	12	7	4	1	12	4	2
S_Count	5	11	5	11	14	12	7	14	5	14	13
ALogP98	5	12	4	7	5	12	13	8	10	7	12
ALogP_MR	15	6	6	1	8	11	2	1	11	15	12
ES_Count_aaaC	4	10	15	2	9	13	3	1	3	1	5
ES_Count_aaCH	7	12	1	10	12	12	4	15	10	9	15
ES_Count_aaN	7	7	5	8	8	14	4	7	11	8	3
ES_Count_aaNH	1	10	7	1	15	11	6	10	13	4	12
ES_Count_aaO	3	5	10	4	7	15	15	2	2	9	10
ES_Count_aasC	9	3	11	1	6	14	10	15	6	3	9
ES_Count_ddsN	11	8	15	14	12	8	11	13	1	5	9
ES_Count_ddssS	2	15	12	8	10	3	14	7	4	12	14
ES_Count_dO	5	4	13	1	14	10	8	1	15	4	11
ES_Count_dsN	15	13	14	3	8	5	8	13	7	12	12
ES_Count_dssC	3	12	5	4	7	11	6	5	2	5	3
ES_Count_sBr	5	3	12	12	2	13	13	10	9	12	10
ES_Count_sCH3	6	7	8	12	12	14	10	1	7	10	13
⋮	⋮			⋮				⋮			⋮
Molecular_Volume	6	13	4	7	6	4	12	7	8	4	1

Lampiran 6 Variabel Prediktor LORENS 25 Subruang *Threshold* 0,5

Variabel	*Ensemble*										
	1	2	3	4	5	6	7	8	9	10	11
pKa.max20.	18	10	2	4	10	11	15	19	23	1	3
Br_Count	3	11	4	7	8	20	11	8	18	18	2
C_Count	1	15	10	14	23	14	6	8	2	15	2
Cl_Count	15	15	23	15	20	22	12	16	25	22	13
F_Count	13	1	13	10	15	16	21	21	9	14	11
H_Count	1	1	13	25	24	21	13	6	12	16	13
I_Count	14	15	8	5	18	11	4	25	4	17	8
N_Count	4	1	15	14	15	5	5	11	2	2	6
O_Count	9	20	21	20	19	11	6	2	20	7	2
S_Count	9	19	8	17	22	19	12	23	7	24	21
ALogP98	8	20	6	11	9	20	21	12	16	12	20
ALogP_MR	25	9	10	1	14	17	4	1	17	24	20
ES_Count_aaaC	6	16	25	4	14	21	5	1	5	2	8
ES_Count_aaCH	11	19	2	17	20	20	7	25	16	14	25
ES_Count_aaN	11	11	9	13	13	23	6	11	18	12	5
ES_Count_aaNH	1	16	12	1	24	18	9	16	22	6	19
ES_Count_aaO	4	7	16	6	11	25	25	4	4	15	16
ES_Count_aasC	15	5	18	2	10	23	16	25	10	5	15
ES_Count_ddsN	18	13	24	24	19	13	18	21	1	7	15
ES_Count_ddssS	3	25	19	13	16	4	23	12	7	20	23
ES_Count_dO	8	6	21	1	22	17	13	1	24	6	17
ES_Count_dsN	24	21	23	5	13	8	13	21	11	19	19
ES_Count_dssC	4	19	7	6	11	18	10	8	4	8	4
ES_Count_sBr	7	5	19	20	4	22	21	16	15	19	17
ES_Count_sCH3	10	12	13	20	19	23	16	2	11	16	22
⋮	⋮			⋮				⋮			⋮
Molecular_Volume	10	21	7	12	9	7	19	11	14	7	2

Lampiran 7 Variabel Prediktor LORENS 30 Subruang *Threshold* 0,5

Variabel	*Ensemble*										
	1	2	3	4	5	6	7	8	9	10	11
pKa.max20.	21	12	2	5	12	13	18	22	28	2	4
Br_Count	3	13	4	8	9	23	13	9	22	22	3
C_Count	1	18	12	17	27	17	7	9	2	18	2
Cl_Count	18	18	27	18	24	27	15	20	30	26	16
F_Count	15	1	16	12	18	20	25	25	11	17	14
H_Count	1	1	15	30	29	25	16	7	14	19	15
I_Count	17	18	9	6	22	14	5	30	5	20	9
N_Count	5	1	18	16	17	6	6	13	2	3	7
O_Count	11	23	25	24	23	13	7	2	24	8	3
S_Count	10	22	10	21	27	23	14	28	9	28	25
ALogP98	10	24	7	13	10	24	25	15	19	14	24
ALogP_MR	30	11	12	1	16	21	4	1	21	29	24
ES_Count_aaaC	8	19	30	4	17	25	6	1	6	2	9
ES_Count_aaCH	14	23	2	20	24	24	8	30	19	17	30
ES_Count_aaN	13	13	10	15	16	27	7	13	21	15	6
ES_Count_aaNH	1	19	14	1	29	21	11	19	26	7	23
ES_Count_aaO	5	9	20	8	13	29	29	4	4	17	20
ES_Count_aasC	18	6	21	2	11	27	19	30	12	6	18
ES_Count_ddsN	22	15	29	28	23	15	21	25	1	9	18
ES_Count_ddssS	4	30	23	16	19	5	27	14	8	24	28
ES_Count_dO	10	7	25	1	27	20	15	1	29	7	21
ES_Count_dsN	29	25	27	6	16	10	15	26	13	23	23
ES_Count_dssC	5	23	9	7	13	21	11	10	4	10	5
ES_Count_sBr	9	6	23	24	4	26	25	19	18	23	20
ES_Count_sCH3	12	14	15	24	23	28	19	2	13	19	26
⋮	⋮			⋮				⋮			⋮
Molecular_Volume	12	25	8	14	11	8	23	14	16	8	2

Lampiran 8. Variabel Prediktor LORENS 45 Subruang *Threshold* 0,5

Variabel	*Ensemble*										
	1	2	3	4	5	6	7	8	9	10	11
pKa.max20.	32	17	3	7	18	19	27	33	41	2	5
Br_Count	5	20	6	11	14	35	19	13	32	33	4
C_Count	1	27	18	25	41	25	10	13	3	27	2
Cl_Count	27	27	41	27	35	40	22	29	45	39	23
F_Count	22	1	24	18	26	29	37	38	16	25	20
H_Count	1	2	22	44	43	37	23	11	20	29	22
I_Count	25	26	13	9	32	20	8	44	7	30	13
N_Count	8	1	27	24	26	9	9	19	3	4	10
O_Count	16	35	37	36	35	19	10	2	36	12	4
S_Count	15	33	15	31	40	34	21	41	13	42	37
ALogP98	14	36	10	19	15	35	37	22	28	20	35
ALogP_MR	45	16	18	2	24	31	6	2	31	43	36
ES_Count_aaaC	11	28	45	6	25	37	9	1	8	3	13
ES_Count_aaCH	20	35	2	30	35	36	12	45	28	25	45
ES_Count_aaN	19	19	15	23	24	40	10	19	32	22	8
ES_Count_aaNH	1	28	21	2	43	32	16	29	39	10	34
ES_Count_aaO	7	13	29	11	19	44	44	6	6	26	29
ES_Count_aasC	27	9	32	3	17	41	28	45	18	8	26
ES_Count_ddsN	32	22	43	42	34	22	31	38	2	13	27
ES_Count_ddssS	5	45	34	24	29	7	40	20	12	36	42
ES_Count_dO	14	10	38	2	40	30	23	2	43	10	31
ES_Count_dsN	43	38	40	9	23	14	23	38	19	34	34
ES_Count_dssC	7	34	13	10	19	32	17	14	6	14	7
ES_Count_sBr	13	9	34	35	6	39	38	29	26	35	30
ES_Count_sCH3	18	21	22	36	34	42	28	3	20	28	39
⋮	⋮			⋮				⋮			⋮
Molecular_Volume	18	37	12	21	16	12	34	20	24	12	3

Lampiran 9. Variabel Prediktor LORENS 50 Subruang *Threshold* 0,5

Variabel	*Ensemble*										
	1	2	3	4	5	6	7	8	9	10	11
pKa.max20.	35	19	3	8	20	21	30	37	46	2	6
Br_Count	5	22	7	13	15	39	21	15	36	36	4
C_Count	1	29	20	28	45	28	12	15	4	30	3
Cl_Count	30	30	45	30	39	44	24	32	50	43	26
F_Count	25	1	26	20	29	32	41	42	18	28	22
H_Count	1	2	25	49	48	41	26	12	23	32	25
I_Count	28	29	15	10	36	22	8	49	7	34	15
N_Count	8	1	29	27	29	10	10	21	3	4	12
O_Count	18	39	41	40	38	21	11	3	40	14	4
S_Count	17	37	16	34	44	38	23	46	14	47	41
ALogP98	16	40	11	22	17	39	41	24	31	23	39
ALogP_MR	50	18	20	2	27	34	7	2	34	48	40
ES_Count_aaaC	12	32	50	7	27	41	10	1	9	4	15
ES_Count_aaCH	22	38	3	33	39	40	13	50	31	28	50
ES_Count_aaN	21	21	17	25	26	45	11	21	35	24	9
ES_Count_aaNH	1	31	23	2	47	35	17	32	43	11	38
ES_Count_aaO	8	14	32	12	21	49	49	7	7	29	32
ES_Count_aasC	30	10	35	3	19	45	31	50	20	9	29
ES_Count_ddsN	36	25	47	47	38	25	35	42	2	14	30
ES_Count_ddssS	6	50	38	26	32	8	45	23	13	40	46
ES_Count_dO	16	11	42	2	44	34	25	2	48	12	34
ES_Count_dsN	47	42	45	10	26	16	25	42	21	38	38
ES_Count_dssC	7	38	14	11	21	35	19	16	7	16	8
ES_Count_sBr	14	10	38	39	7	43	42	32	29	38	34
ES_Count_sCH3	19	23	25	40	37	46	32	3	22	31	44
⋮											
Molecular_Volume	20	41	13	24	18	13	38	22	27	13	3

Lampiran 10 Probabilitas Akhir LORENS 5 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.000601	0.2	3.06E-06	⋯	0.2
2	7.16E-34	1.15E-12	3.08E-12		2.16E-23
3	4.69E-17	2.67E-11	2.14E-21		5.85E-23
4	3.58E-14	0.2	2.68E-12		4.15E-05
5	0.200082	0.2	1.67E-21		0.6
6	0.193321	0.276262	0.618253		0.386402
7	0.2	2.38E-32	8.39E-07		0.2
8	0.8	0.600001	0.795526	⋯	0.4
9	1.28E-38	0.00534	4.55E-63		0.2
10	8.11E-22	0.199465	1.22E-52		8.94E-23
11	0.000172	0.400735	0.2		0.2
12	0.04801	0.200021	0.6		0.4
13	0.401959	0.399999	0.400482		0.6
14	0.215786	0.4	0.4		0.2
15	0.6	0.6	0.6		0.225894
16	0.399983	0.395689	0.4		0.270724
17	0.400271	0.400111	0.6	⋯	0.2

Lampiran 11 Probabilitas Akhir LORENS 8 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.125	1.58E-42	0.25	⋯	3.05E-101
2	0.125	0.124999	0.25		0.125
3	2.00E-10	9.51E-08	4.59E-20		1.16E-39
4	1.73E-20	0.125	3.96E-28		2.22E-23
5	0.26608	0.125	0.125		0.25
6	0.499499	0.5	0.750002		0.625
7	0.125975	0.12499	0.125		0.250269
8	0.250005	0.375	0.5	⋯	0.375
9	2.81E-57	8.33E-88	1.95E-23		5.01E-39
10	1.27E-23	1.76E-76	1.12E-34		0.125
11	0.25	0.125	0.374969		0.25
12	0.25	0.249995	0.375		0.25
13	0.125047	0.252519	0.522196		0.131801
14	0.125	0.25	0.125		2.66E-38
15	0.402666	0.125	0.25		0.25
16	0.624618	0.375	0.375		0.5
17	0.5	0.25	0.625	⋯	0.492991

Lampiran 12 Probabilitas Akhir LORENS 10 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	7.00E-33	1.10E-20	2.15E-07	⋯	2.56E-10
2	1.71E-06	0.076007	0.286669		0.100854
3	0.1	0.171611	0.025845		0.300588
4	0.1	0.100356	0.000445		0.000508
5	0.200596	0.3	0.445773		0.1
6	0.541281	0.653671	0.76091		0.691225
7	0.1	0.397241	2.74E-05		0.4
8	0.406277	0.361028	0.383379	⋯	0.372455
9	2.66E-18	4.39E-14	0.002061		1.80E-15
10	6.41E-11	0.1	4.73E-05		3.08E-15
11	0.300001	0.200014	0.20963		0.203884
12	0.2	0.504698	0.11575		0.38396
13	0.203941	0.200022	0.505194		0.1
14	0.1	0.2	0.2		0.090194
15	0.2	0.2	0.1		0.1
16	0.610793	0.392158	0.223939		0.300267
17	0.283645	0.3	0.2	⋯	0.1

Lampiran 13 Probabilitas Akhir LORENS 15 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.005609	0.001102	0.027054	⋯	0.003888
2	0.217741	0.202841	0.196138		0.173824
3	0.11428	0.090251	0.187867		0.149204
4	0.05496	0.046932	0.063403		0.072738
5	0.208669	0.092158	0.147883		0.206883
6	0.340151	0.398448	0.468131		0.28089
7	0.201659	0.112708	0.03796		0.087153
8	0.391446	0.260616	0.277417	⋯	0.379266
9	0.019345	0.044423	0.056589		0.070069
10	0.004596	0.006572	0.011977		0.027399
11	0.307926	0.231265	0.195433		0.323476
12	0.058343	0.119683	0.135065		0.189958
13	0.285086	0.148979	0.207143		0.362452
14	0.022365	0.00577	0.095738		0.011875
15	0.000263	0.000486	0.000802		0.000234
16	0.305567	0.362592	0.328947		0.400448
17	0.009867	0.069737	0.000236	⋯	1.36E-05

Lampiran 14 Probabilitas Akhir LORENS 20 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.002805	0.002208	0.007402	⋯	0.010205
2	0.204729	0.178834	0.20222		0.187769
3	0.115662	0.133978	0.135389		0.122373
4	0.072174	0.061211	0.070635		0.073338
5	0.162963	0.142495	0.135954		0.16816
6	0.322542	0.311226	0.265143		0.303986
7	0.053741	0.070621	0.061216		0.076935
8	0.227523	0.274768	0.220433	⋯	0.280862
9	0.082459	0.052615	0.052097		0.070748
10	0.033304	0.031175	0.039285		0.042947
11	0.227594	0.275646	0.276358		0.284139
12	0.06026	0.175058	0.136324		0.200033
13	0.290519	0.246456	0.266189		0.306118
14	0.011518	0.005159	0.062107		0.006707
15	0.014943	0.00054	0.024821		0.009777
16	0.292147	0.272565	0.250294		0.302931
17	0.003937	0.014828	0.031451	⋯	0.002352

Lampiran 15 Probabilitas Akhir LORENS 25 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.005341	0.00579	0.012897	⋯	0.015617
2	0.197705	0.196066	0.228079		0.156861
3	0.127486	0.110524	0.110471		0.109612
4	0.088642	0.061353	0.076135		0.068254
5	0.119327	0.086003	0.096647		0.110966
6	0.24555	0.228124	0.178782		0.19703
7	0.047838	0.030466	0.055749		0.035916
8	0.179884	0.180174	0.16494	⋯	0.157135
9	0.0573	0.036264	0.066736		0.045278
10	0.036192	0.025011	0.037175		0.031546
11	0.264784	0.252546	0.241023		0.214979
12	0.126739	0.158686	0.173508		0.151706
13	0.228771	0.225532	0.237787		0.273961
14	0.011472	0.003126	0.020733		0.007028
15	0.017679	0.006892	0.009803		0.008879
16	0.225592	0.207364	0.194882		0.197879
17	0.037189	0.021947	0.012481	⋯	0.007676

Lampiran 16 Probabilitas Akhir LORENS 30 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.011555	0.010653	0.010219	⋯	0.00853
2	0.184945	0.199972	0.198554		0.19935
3	0.105666	0.106009	0.106133		0.111774
4	0.079901	0.076652	0.073936		0.086232
5	0.133137	0.099936	0.086784		0.122844
6	0.172707	0.155288	0.200195		0.167512
7	0.025042	0.065841	0.057731		0.027654
8	0.17818	0.134902	0.141017	⋯	0.161189
9	0.070554	0.042387	0.053353		0.052613
10	0.051282	0.034914	0.03948		0.042167
11	0.24406	0.250454	0.197		0.230716
12	0.133763	0.143348	0.1702		0.144041
13	0.269762	0.255997	0.212931		0.229036
14	0.010101	0.011402	0.011609		0.009677
15	0.012349	0.012913	0.014304		0.01453
16	0.195002	0.142454	0.204618		0.173223
17	0.005292	0.020425	0.007625	⋯	0.012386

Lampiran 17 Probabilitas Akhir LORENS 40 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.01133	0.017485	0.014029	⋯	0.013657
2	0.186543	0.199079	0.183729		0.196766
3	0.094307	0.09203	0.086986		0.105054
4	0.078239	0.073012	0.071685		0.084694
5	0.089618	0.108819	0.084004		0.094098
6	0.135939	0.13247	0.126992		0.135062
7	0.033336	0.031247	0.028748		0.03277
8	0.113206	0.130599	0.111919	⋯	0.122024
9	0.064835	0.055859	0.065897		0.061788
10	0.051455	0.047091	0.051144		0.048372
11	0.156405	0.185102	0.179099		0.191955
12	0.171681	0.169198	0.14945		0.162299
13	0.211889	0.249325	0.227107		0.229418
14	0.017203	0.015618	0.012414		0.015077
15	0.018875	0.019896	0.019504		0.019656
16	0.126276	0.136337	0.140822		0.144229
17	0.011903	0.014798	0.009226	⋯	0.00859

Lampiran 18 Probabilitas Akhir LORENS 45 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.016442	0.017067	0.022771	⋯	0.018057
2	0.198713	0.198196	0.179244		0.19464
3	0.099517	0.095085	0.087049		0.090792
4	0.086828	0.081442	0.074582		0.076605
5	0.091855	0.080248	0.093768		0.082966
6	0.12345	0.122946	0.115355		0.123042
7	0.033135	0.031776	0.034447		0.03735
8	0.117751	0.112685	0.110442	⋯	0.107646
9	0.065169	0.057403	0.062835		0.063449
10	0.054284	0.048715	0.054685		0.051784
11	0.156386	0.174762	0.152263		0.162705
12	0.169849	0.173289	0.172152		0.173296
13	0.207617	0.202173	0.224246		0.180039
14	0.018066	0.012247	0.015304		0.019942
15	0.022027	0.014689	0.019929		0.022486
16	0.125465	0.128535	0.122505		0.131166
17	0.013285	0.010186	0.014347	⋯	0.016336

Lampiran 19 Probabilitas Akhir LORENS 50 Subruang *Threshold* 0,5

No	*Ensemble*				
	1	2	3		11
1	0.017073	0.020942	0.019961	⋯	0.02189
2	0.201021	0.201798	0.182175		0.186144
3	0.097196	0.095984	0.082234		0.090699
4	0.084153	0.083049	0.07015		0.080551
5	0.086859	0.081795	0.085309		0.085562
6	0.125016	0.121876	0.111821		0.119075
7	0.033849	0.039077	0.032523		0.043889
8	0.106053	0.105016	0.10503	⋯	0.110318
9	0.06712	0.059496	0.066693		0.063342
10	0.056941	0.051349	0.053496		0.054551
11	0.16997	0.184171	0.163708		0.160594
12	0.168977	0.176676	0.171476		0.169027
13	0.195999	0.195091	0.214449		0.183207
14	0.017578	0.022723	0.016831		0.024119
15	0.021096	0.019829	0.02429		0.026499
16	0.113007	0.11737	0.121229		0.123783
17	0.014924	0.016442	0.012258	⋯	0.016001

Lampiran 20 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Pertama

No	Probability	Prediksi
1	0,0284	0
2	0,1702	0
3	0,1096	0
4	0,0992	0
5	0,1094	0
6	0,1198	0
7	0,0617	0
8	0,1095	0
9	0,0994	0
10	0,0892	0
11	0,0793	0
12	0,1783	0
13	0,1701	0
14	0,0276	0
15	0,0451	0
16	0.1201	0
17	0.0159	0

Lampiran 21 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Kedua

No	Probability	Prediksi
1	0.0278	0
2	0.1800	0
3	0.0919	0
4	0.0836	0
5	0.1001	0
6	0.1004	0
7	0.0469	0
8	0.1002	0
9	0.0877	0
10	0.0832	0
11	0.0827	0
12	0.1940	0
13	0.1703	0
14	0.0210	0
15	0.0366	0
16	0.0868	0
17	0.0212	0

Lampiran 22 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Ketiga

No	Probability	Prediksi
1	0.0207	0
2	0.1869	0
3	0.0946	0
4	0.0839	0
5	0.0967	0
6	0.1054	0
7	0.0445	0
8	0.0981	0
9	0.0840	0
10	0.0768	0
11	0.0737	0
12	0.2030	0
13	0.1744	0
14	0.0170	0
15	0.0319	0
16	0.0942	0
17	0.0127	0

Lampiran 23 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Keempat

No	Probability	Prediksi
1	0.0199	0
2	0.1859	0
3	0.0970	0
4	0.0859	0
5	0.0995	0
6	0.1085	0
7	0.0457	0
8	0.1002	0
9	0.0868	0
10	0.0782	0
11	0.0728	0
12	0.2003	0
13	0.1788	0
14	0.0172	0
15	0.0324	0
16	0.1008	0
17	0.0097	0

Lampiran 24 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Kelima

No	Probability	Prediksi
1	0.0245	0
2	0.1799	0
3	0.1074	0
4	0.0961	0
5	0.1038	0
6	0.1177	0
7	0.0546	0
8	0.1052	0
9	0.0948	0
10	0.0840	0
11	0.0808	0
12	0.1845	0
13	0.1756	0
14	0.0210	0
15	0.0374	0
16	0.1176	0
17	0.0127	0

Lampiran 25 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Keenam

No	Probability	Prediksi
1	0.1045	0
2	0.1045	0
3	0.1045	0
4	0.1045	0
5	0.1045	0
6	0.1045	0
7	0.1045	0
8	0.1045	0
9	0.1045	0
10	0.1045	0
11	0.1045	0
12	0.1045	0
13	0.1045	0
14	0.1045	0
15	0.1045	0
16	0.1045	0
17	0.1045	0

Lampiran 26 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Ketujuh

No	Probability	Prediksi
1	0.1045	0
2	0.1045	0
3	0.1045	0
4	0.1045	0
5	0.1045	0
6	0.1045	0
7	0.1045	0
8	0.1045	0
9	0.1045	0
10	0.1045	0
11	0.1045	0
12	0.1045	0
13	0.1045	0
14	0.1045	0
15	0.1045	0
16	0.1045	0
17	0.1045	0

Lampiran 27 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Kedelapan

No	Probability	Prediksi
1	0.0205	0
2	0.1835	0
3	0.1050	0
4	0.0932	0
5	0.1036	0
6	0.1167	0
7	0.0507	0
8	0.1042	0
9	0.0930	0
10	0.0816	0
11	0.0762	0
12	0.1906	0
13	0.1814	0
14	0.0188	0
15	0.0349	0
16	0.1169	0
17	0.0082	0

Lampiran 28 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Kesembilan

No	Probability	Prediksi
1	0.0304	0
2	0.1693	0
3	0.1096	0
4	0.1001	0
5	0.1093	0
6	0.1189	0
7	0.0619	0
8	0.1095	0
9	0.1005	0
10	0.0909	0
11	0.0869	0
12	0.1740	0
13	0.1691	0
14	0.0284	0
15	0.0463	0
16	0.1195	0
17	0.0180	0

Lampiran 29 Prediksi Probabilitas Metode ELR Model Terbaik Pengulangan Kesepuluh

No	Probability	Prediksi
1	0.0148	0
2	0.2019	0
3	0.1012	0
4	0.0891	0
5	0.0933	0
6	0.1112	0
7	0.0381	0
8	0.0950	0
9	0.0866	0
10	0.0740	0
11	0.0876	0
12	0.1951	0
13	0.1958	0
14	0.0095	0
15	0.0230	0
16	0.1112	0
17	0.0046	0

Lampiran 30 *Syntax Synthetic Minority Oversampling Thecnique* (SMOTE)

```
library(UBL)
data<-read.csv("C:/Users/Charles/Documents/Kuliah/Semester
8/Tugas Akhir/File/Data TA/Dataa.csv", header = TRUE, sep =
";")
train <- read.csv("C:/Users/Charles/Documents/Kuliah/Semester
8/Tugas Akhir/File/Data TA/train.csv", header = TRUE, sep =
",")[,-c(1)]
test <- read.csv("C:/Users/Charles/Documents/Kuliah/Semester
8/Tugas Akhir/File/Data TA/test.csv", header = TRUE, sep =
",")[,-c(1)]

for (i in 1:nrow(train)){
  Y <- ifelse(train$Y==1, "Pos", "Neg")
}
newtrain <- data.frame(Y, train[,-c(1)])

trainSMOTEBalance <- SmoteClassif(Y ~ ., newtrain, C.perc =
list(Neg = 1,Pos = 8.571429))
table(trainSMOTEBalance$Y)

for (i in 1:nrow(trainSMOTEBalance)){
  Y <- ifelse(trainSMOTEBalance$Y=="Pos", 1, 0)
}
train <- data.frame(Y, trainSMOTEBalance[,-c(1)])
```

Lampiran 31. *Syntax Logistic Regression Ensembles* (LORENS)

```
lr.cerp <-
function(y,x,nens,fixsize=NULL,fixthres=NULL,search=F) {

# initialization
set.seed(as.numeric(Sys.time()))
options(warn=-1)
if(sum(is.na(x))>0) stop("missing value is found")
if(sum(is.na(y))>0) stop("missing value is found")
y <- as.data.frame(y)
x <- as.data.frame(x)
num_pred <- ncol(x)
num_obs <- nrow(x)
pos_rate <- sum(y)/num_obs

# parameter search or default option
if(search==T) {
    optimal <- search.thre_size(y,x,"lr")
    optsize <- optimal$size; opthreshold <- optimal$threshold
    }
else {
    if(is.null(fixsize)) fixsize <-round(6*num_pred/num_obs)
    if(is.null(fixthres)) fixthres <- (pos_rate+.5)/2
    optsize <- fixsize; opthreshold <- fixthres
    }

# main body
ptss <- floor(seq(1,optsize+.999,length.out=num_pred))
fitted <- NULL; predicted <- NULL; cname <- NULL;
coef.table<-matrix(0,num_pred,nens);
partition.table <- matrix(0,num_pred,nens); intc <-
matrix(0,optsize,nens); probability <- rep(0,num_obs)
for (i in 1:nens) {
cname <- c(cname, paste("ens",i,sep=""))
rand_pred <- sample(ptss)
partition.table[,i] <- rand_pred
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
avg_fit <- rep(0,num_obs)
for(j in 1:optsize) {
        smp_dt <- cbind(y,x[,rand_pred==j])
        intlr <- glm(y~.,data=smp_dt,family=binomial())
        coef.vector <- intlr$coefficient
        coef.vector[is.na(coef.vector)] <- 0
        intc[j,i] <- coef.vector[1]; coef.vector <- coef.vector[-1]
                        coef.table[rand_pred==j,i] <- coef.vector
                        avg_fit <- avg_fit + intlr$fitted.values
                }
fitted <- cbind(fitted,avg_fit/optsize,deparse.level=0)
                probability <- probability+(avg_fit/optsize)/nens
        }
learning.decision <- ens.voting(fitted,opthreshold)$final.vote
        colnames(fitted) <- cname
        colnames(intc) <- cname
        colnames(coef.table) <- cname; rownames(coef.table) <-
colnames(x)
colnames(partition.table) <- cname; rownames(partition.table) <-
colnames(x)

return(list(fitted=fitted,probability=probability,learning.decision=
learning.decision,

partition.table=partition.table,coef.table=coef.table,intercept=intc,

number.ensemble=nens,optimal.size=optsize,optimal.threshold=o
pthreshold))
}

### lr.cerp.predict applies lr.cerp model to new data(test set)
similar as predict.lm function.
### lr.cerp.object is required and built from lr.cerp function.
### xtest is also required and should be same format as x in
lr.cerp function.
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
### ytest is optional if you want to check the accuracy
lr.cerp.predict <- function(lr.cerp.object,xtest,ytest=NULL) {
# initialization
options(warn=-1)
if(sum(is.na(xtest))>0) stop("missing value is found")
if(sum(is.na(ytest))>0) stop("missing value is found")
xtest <- as.data.frame(xtest)
num_obs <- nrow(xtest)
nens <- lr.cerp.object$number.ensemble
optsize <- lr.cerp.object$optimal.size
opthreshold <- lr.cerp.object$optimal.threshold

# main body
cname <- NULL; test.decision <- NULL; fitted <- NULL;
probability <- rep(0,num_obs)
xtest <- xtest[,rownames(lr.cerp.object$partition.table)]
for (i in 1:nens) {
        avg_fit <- rep(0,num_obs)
        cname <- c(cname, paste("ens",i,sep=""))
        curmod <- lr.cerp.object$partition.table[,i]
        for(j in 1:optsize) {
                intc <- lr.cerp.object$intercept[j,i]
                wrkmat <- xtest[,curmod==j]
                cvec <- lr.cerp.object$coef.table[curmod==j,i]
                        int_vl <- as.matrix(wrkmat)%*%cvec
                        int_vl <- int_vl + intc
                        int_vl[int_vl>=709] <- 709
                        avg_fit <- avg_fit +
exp(int_vl)/(1+exp(int_vl))
                }
        fitted <- cbind(fitted,avg_fit/optsize,deparse.level=0)
                probability <- probability+(avg_fit/optsize)/nens
        }
test.decision <- ens.voting(fitted,opthreshold,ytest)
colnames(fitted) <- cname
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
return(list(fitted=fitted,probability=t(probability),
decision=test.decision$final.vote,
optimal.size=optsize,optimal.threshold=opthreshold,decision.tabl
e=test.decision$twobytwo))
}

### lr.cerp.cv performs v-fold cross-validation using lr.cerp and
lr.cerp.predict functions.
### Options and requirements are the same as lr.cerp function.
### One additional requirement is v_fold which is the number of
fold to be performed for cross-validation.
lr.cerp.cv <-
function(y,x,nens,v_fold,fixsize=NULL,fixthres=NULL,search=F
) {
# initialization
set.seed(as.numeric(Sys.time()))
options(warn=-1)
if(sum(is.na(x))>0) stop("missing value is found")
if(sum(is.na(y))>0) stop("missing value is found")
y <- as.data.frame(y)
x <- as.data.frame(x)
num_obs <- nrow(y)
rand_obs <- sample(1:num_obs)
obs_rem <- num_obs%%v_fold
obs_div <- (num_obs-obs_rem)/v_fold

# main body
probability <- rep(0,num_obs); predicted <- rep(0,num_obs);
tbttable <- matrix(0,2,2)
part_size.list<-NULL; threshold.list<-NULL
for(i in 1:v_fold) {
        if(i<=obs_rem) {head1<-(i-1)*(obs_div+1)+1;tail1<-
i*(obs_div+1);}
        else {head1<-(i-1)*obs_div+obs_rem+1;tail1<-
i*obs_div+obs_rem;}
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
test_seq<-rand_obs[head1:tail1]
        learn_seq<-rand_obs[-c(head1:tail1)]
        ylearn<-y[learn_seq,];xlearn<-x[learn_seq,];xtest<-
x[test_seq,];ytest<-y[test_seq,]
mid_rs<-lr.cerp(ylearn,xlearn,nens,fixsize,fixthres,search)
                pred_rs<-lr.cerp.predict(mid_rs,xtest,ytest)
                predicted[test_seq]<-pred_rs$decision
for(j in 1:nens) probability[test_seq]<-
probability[test_seq]+pred_rs$fitted[,j]/nens
        tbttable<-tbttable+pred_rs$decision.table
        part_size.list<-c(part_size.list,mid_rs$optimal.size)
        threshold.list<-c(threshold.list,mid_rs$optimal.threshold)
        }

return(list(probability=probability,predicted=predicted,partition.si
ze.list=part_size.list,
        threshold.list=threshold.list,decision.table=tbttable))
}

### internal functions
ens.voting <- function (tot_res,threshold,y=NULL) {
        nens<-ncol(tot_res);nobs<-nrow(tot_res)
        if (!is.null(y)) {real_pos<-sum(y);real_neg<-nobs-
real_pos}
        tot_res[tot_res>=threshold] <- 1;
tot_res[tot_res<threshold] <- 0
        final.vote <- rep(0,nobs)
        for(i in 1:nobs) final.vote[i] <- mean(tot_res[i,])
        final.vote[final.vote>=0.5] <- 1; final.vote[final.vote<0.5]
<- 0
        twobytwo <- NULL
        if (!is.null(y)) {
                real_pred_pos <- sum(final.vote==y&y==1)
                real_pred_neg <- sum(final.vote==y&y==0)
                real_pos_pred_neg <- real_pos - real_pred_pos
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
real_neg_pred_pos <- real_neg - real_pred_neg
                twobytwo <-
rbind(c(real_pred_pos,real_pos_pred_neg),c(real_neg_pred_pos,r
eal_pred_neg))
                rownames(twobytwo) <- c("real.pos","real.neg")
                colnames(twobytwo) <- c("pred.pos","pred.neg")
        }
        return(list(final.vote=final.vote,twobytwo=twobytwo))
}
search.thre_size <- function (y,x,method) {
        nprd <- ncol(x);nobs <- nrow(x);orate <- sum(y)/nobs
        szseq <- NULL; int_fits <- NULL
        initseed <- c(2,3,4,5,6,7,8,9,10,12)
        for (i in initseed) {
                ipts<-i*nprd/nobs
                ipts<-floor(ipts)
                if (ipts%%2==0) ipts<-ipts+1
                if (szseq[length(szseq)]!=ipts||is.null(szseq)) {
                        szseq <- c(szseq,ipts)
                        int_fits <-
cbind(int_fits,cv.fit(y,x,ipts,method))
                }
        }
        nsrsz <- length(szseq)
        add_fits<-NULL;addsz<-NULL
        if(orate>=.5) iseq<-seq(.5,orate,.02)
        else {iseq<-seq(.5,orate,-.02); iseq<-rev(iseq)}
        nbis<-length(iseq)
        szfth<-rep(0,nbis);acfth<-rep(0,nbis)
        for(j in 1:nbis) {
                acseq<-rep(0,nsrsz)
                for(k in 1:nsrsz) {
                        tmpf<-rep(0,nobs)
                        tmpf[int_fits[,k]>=iseq[j]]<-
1;tmpf[int_fits[,k]<iseq[j]]<-0
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
acseq[k]<-sum(tmpf==y)/nobs
                }
                nbst<-sum(acseq==max(acseq));scol<-
seq(1:nsrsz)
                if(nbst==1) nthc<-scol[acseq==max(acseq)]
                else {
                        tmpcol<-scol[acseq==max(acseq)]
                        nthc<-tmpcol[round(nbst/2)]
                }
                if(nthc==1) {
                        upts<-szseq[nthc+1];lpts<-szseq[nthc]
                        utfac<-acseq[nthc+1];ltfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
                                mtf<-cv.fit(y,x,mpts,method)
                                addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
                                else if(sum(addsz==mpts)==0) {
                                        mtf<-
cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else mtf<-
add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-
1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
        if(mtfac>=utfac) {upts<-mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-
mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-
utfac}
                                }
                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-
mpts;ltfac<-mtfac
                                                upts<-
mpts;utfac<-mtfac
                                        }
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                        }
                        if(ltfac>=utfac) {szfth[j]<-lpts;acfth[j]<-
ltfac}
                        else {szfth[j]<-upts;acfth[j]<-utfac}
                }
                else if(nthc==nsrsz) {
                        lpts<-szseq[nthc-1];upts<-szseq[nthc]
                        ltfac<-acseq[nthc-1];utfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
mtf<- cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else if(sum(addsz==mpts)==0) {
                                        mtf<-
cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else mtf<-
add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-
1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
                                        if(mtfac>=utfac) {upts<-
mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-
mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-
utfac}
                                }
                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-
mpts;ltfac<-mtfac
                                                upts<-
mpts;utfac<-mtfac
                                        }
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
                else {upts<-lpts;utfac<-ltfac}
                                }
                        }
                        if(ltfac>=utfac) {szfth[j]<-lpts;acfth[j]<-
ltfac}
                        else {szfth[j]<-upts;acfth[j]<-utfac}
                }
                else {
                        lpts<-szseq[nthc-1];upts<-szseq[nthc]
                        ltfac<-acseq[nthc-1];utfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
                                        mtf<-
cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else if(sum(addsz==mpts)==0) {
                                        mtf<-
cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else mtf<-
add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-
1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
                                        if(mtfac>=utfac)
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
{upts<- mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-
mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-
utfac}
                                }
                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-
mpts;ltfac<-mtfac
                                                upts<-
mpts;utfac<-mtfac
                                        }
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                        }
                        if(ltfac>=utfac) {lsps<-lpts;lsbs<-ltfac}
                        else {lsps<-upts;lsbs<-utfac}
                        upts<-szseq[nthc+1];lpts<-szseq[nthc]
                        utfac<-acseq[nthc+1];ltfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
                                        mtf<-
cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf) }
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
else if(sum(addsz==mpts)==0) {
                                        mtf<-
cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else mtf<-
add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-
1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
                                        if(mtfac>=utfac) {upts<-
mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-
mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-
utfac}
                                }
                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-
mpts;ltfac<-mtfac
                                                upts<-
mpts;utfac<-mtfac
                                        }
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                        }
```

Lampiran 31 *Syntax Logistic Regression Ensembles* (LORENS) (Lanjutan)

```
if(ltfac>=utfac) {usps<-lpts;usbs<-ltfac}
                        else {usps<-upts;usbs<-utfac}
                        if(lsbs>=usbs) {szfth[j]<-lsps;acfth[j]<-
lsbs}
                        else {szfth[j]<-usps;acfth[j]<-usbs}
                }
        }
        fnbst<-sum(max(acfth)==acfth);fscol<-seq(1:nbis)
        if(fnbst==1) {
                finsz<-szfth[max(acfth)==acfth]
                finth<-iseq[max(acfth)==acfth]
        }
        else {
                ftmpcol<-fscol[max(acfth)==acfth]
                tgcol<-ftmpcol[round(fnbst/2)]
                finsz<-szfth[tgcol]
                finth<-iseq[tgcol]
        }
        return(list(size=finsz,threshold=finth))
}

cv.fit <- function (y,x,npt,method) {
        num_pred<-ncol(x)
        num_obs<-nrow(x)
        lfit<-rep(0,num_obs)
        nv=3
        if(method=="lr") lfit<-lr.cerp.cv(y,x,1,nv)$probability
        else if(method=="lrt") lfit<-
lrt.cerp.cv(y,x,1,nv)$probability
        else if(method=="ct") lfit<-
ct.cerp.cv(y,x,1,nv)$probability
        return(lfit)
}
```

Lampiran 32 *Syntax Ensemble Logistic Regression* (ELR)

```
base.model = function(ytrain, xtrain, ytest = NULL, xtest =
NULL, alpha = 0, fix.lambda = NULL, nlambda = 100,
type.measure = "auc", nfolds.reg.lr = 3) {
   require(glmnet)
   if (is.null(fix.lambda)) {
      repeat {
         model = try(cv.glmnet(y = as.factor(ytrain), x =
as.matrix(xtrain), family = "binomial", alpha = alpha, nlambda =
nlambda,
                type.measure = type.measure, nfolds = nfolds.reg.lr,
standardize = FALSE),silent = TRUE)
         if (isTRUE(class(model) != "try-error")) break
         else print("Please wait...")
      }
      lambda=model$lambda.min
   }
   else {
      repeat {
         model = try(glmnet(y = as.factor(ytrain), x =
as.matrix(xtrain), family = "binomial", alpha = alpha, lambda =
fix.lambda,
                standardize = FALSE), silent = TRUE)
         if (isTRUE(class(model) != "try-error")) break
         else print("Please wait...")
      }
      lambda=model$lambda
   }

   coef = coef(model, s = "lambda.min")
   train.pred = as.factor(predict(model, as.matrix(xtrain), s =
"lambda.min", type = "class"))
   levels(train.pred) =levels(as.factor(ytrain))
   train.prob = predict(model, as.matrix(xtrain), s =
"lambda.min", type = "response")
   test.pred = as.factor(predict(model, as.matrix(xtest), s =
"lambda.min", type = "class"))
```

Lampiran 32 *Syntax Ensemble Logistic Regression* (ELR) (Lanjutan)

```
    levels(test.pred)=levels(as.factor(ytest))
    test.prob = predict(model, as.matrix(xtest), s = "lambda.min",
type = "response")

    require(pROC)
    require(caret)
    conf.train = confusionMatrix(data = as.factor(train.pred),
reference = as.factor(ytrain))
    acc.train = conf.train$overall[1]
    sensitivity.train = conf.train$byClass[2]
    specificity.train = conf.train$byClass[1]
    bcr.train = conf.train$byClass[11]
    auc.train = pROC::roc(as.factor(ytrain),
as.vector(as.numeric(train.prob)))$auc

    conf.test = confusionMatrix(data = as.factor(test.pred),
reference = as.factor(ytest))
    acc.test = conf.test$overall[1]
    sensitivity.test = conf.test$byClass[2]
    specificity.test = conf.test$byClass[1]
    bcr.test = conf.test$byClass[11]
    auc.test = pROC::roc(as.factor(ytest),
as.vector(as.numeric(test.prob)))$auc

    return(list(model = model, coefficients = coef, training.data =
cbind(y = ytrain, xtrain), testing.data = cbind(y = ytest, xtest),
    train.prediction = train.pred, train.probability = train.prob,
test.prediction = test.pred, test.probability = test.prob, conf.train =
conf.train, conf.test = conf.test,
    bcr = c(training.bcr = bcr.train, testing.bcr = bcr.test), accuracy
= c(training.accuracy = acc.train, testing.accuracy = acc.test),
    sensitivity = c(training.sensitivity = sensitivity.train,
testing.sensitivity = sensitivity.test),
    specificity = c(training.specificity = specificity.train,
testing.specificity = specificity.test),
```

Lampiran 32 *Syntax Ensemble Logistic Regression* (ELR) (Lanjutan)

```
    auc = c(training.AUC = auc.train, testing.AUC = auc.test),
    lambda = lambda))
}

elr = function(ytrain, xtrain, ytest, xtest, prob.feature, tr.ratio = 8 /
10, alpha = 0, fix.lambda = NULL, nlambda = 100, type.measure
= "auc", nfolds.reg.lr = 3, tol = 1e-5) {
    n.feature = ncol(xtrain)

    bcr.value = acc.value = sensitivity.value = specificity.value =
auc.value = NULL
    quality = NULL
    avg.bcr = NULL
    cfeat = NULL
    coefis = NULL
    conf.train = NULL
    conf.test = NULL
    prob.baru = NULL
    lambda.iterasi.min = NULL
    train.pred.iter = train.prob.iter = NULL
    test.pred.iter = test.prob.iter = NULL
    a = b = c = d = e = f = g = h = NULL
    avg.bcr[1] = (length(which(y.train == 1)) + length(which(y.test
== 1)))/(length(y.train)+length(y.test))
    prob = matrix(nrow = length(prob.feature))
    prob[, 1] = prob.feature

    iter = 1
    bcr.value = auc.value = sensitivity.value = specificity.value =
acc.value = feature = NULL
    repeat {
        iter = iter + 1
        cat("iter", iter, "\n")
        repeat {
            ytrain = ytrain
```

Lampiran 32 *Syntax Ensemble Logistic Regression* (ELR) (Lanjutan)

```
        xtrain = xtrain
        ytest = ytest
        xtest = xtest

        prob_tiap_iterasi= prob[, iter-1]
        chs.feature = sort(sample(x = c(1:n.feature), size =
nrow(xtrain), replace = FALSE, prob = prob[, iter - 1]))

        dtrain = xtrain[, chs.feature]
        dtest = xtest[, chs.feature]
        ttrain = ytrain
        ttest = ytest

        model = base.model(ytrain = as.vector(ttrain), xtrain =
as.matrix(dtrain),
                ytest = as.vector(ttest), xtest = as.matrix(dtest), alpha
= alpha, fix.lambda=NULL, nlambda = nlambda,
                type.measure = type.measure, nfolds.reg.lr =
nfolds.reg.lr)

        coef.model = coef(model, s = "lambda.min")
        coef.m = coef.model
        coefi = as.vector(coef.model)
        cfeat = chs.feature
        lambda = model$lambda
        lambdamin = model$lambda
        train.pred = model$train.prediction
        train.prob = model$train.probability
        test.pred = model$test.prediction
        test.prob = model$test.probability

        quality[iter] = log10(1 + abs(model$bcr[2] - avg.bcr[iter -
1]))
        upd.prob = prob[chs.feature] + quality[iter] *
as.vector(coef.model)[-1] ^ (2 * sign(quality[iter]))
```

Lampiran 32 *Syntax Ensemble Logistic Regression* (ELR) (Lanjutan)

```
upd.prob[which(upd.prob == -Inf | upd.prob == Inf)] = 0
          prob = cbind(prob, rep(NA, length(prob.feature)))
          prob[chs.feature, iter] = abs(upd.prob) /
abs(upd.prob[which.max(abs(upd.prob))[1]])
          prob[-chs.feature, iter] = prob[-chs.feature, (iter - 1)]
          avg.bcr[iter] = (((iter - 1) * avg.bcr[iter - 1]) +
model$bcr[2]) / iter

          if (length(which(prob[, iter] < 0)) == 0) break
          }
       bcr.value = rbind(bcr.value, model$bcr)
       acc.value = rbind(acc.value, model$accuracy)
       sensitivity.value = rbind(sensitivity.value, model$sensitivity)
       specificity.value = rbind(specificity.value,
model$specificity)
       auc.value = rbind(auc.value, model$auc)
       feature=cbind(feature,cfeat)
       d = model$conf.train$table[[1]]
       c = model$conf.train$table[[2]]
       b = model$conf.train$table[[3]]
       a = model$conf.train$table[[4]]
       h = model$conf.test$table[[1]]
       g = model$conf.test$table[[2]]
       f = model$conf.test$table[[3]]
       e = model$conf.test$table[[4]]
       train.conf = cbind(a,b,c,d)
       test.conf = cbind(e,f,g,h)
       conf.train = rbind( conf.train, train.conf)
       conf.test = rbind(conf.test, test.conf)
       prob.baru = cbind(prob.baru, prob_tiap_iterasi)
       coefis = cbind(coefis, coefi)
       lambda.iterasi.min = rbind(lambda.iterasi.min, lambdamin)
       train.pred.iter = cbind(train.pred.iter, train.pred)
       train.prob.iter = cbind(train.prob.iter, train.prob)
       test.pred.iter = cbind(test.pred.iter, test.pred)
```

Lampiran 32 *Syntax Ensemble Logistic Regression* (ELR) (Lanjutan)

```
test.prob.iter = cbind(test.prob.iter, test.prob)

      eps = abs(avg.bcr[iter - 1] - avg.bcr[iter])
      if (eps <= tol & iter > 10) break
      }

   return(list(quality=quality, upd.prob=prob,
prob.baru=prob.baru, conf.train = conf.train, conf.test = conf.test,
model = model, bcr = bcr.value, accuracy = acc.value, sensitivity
= sensitivity.value,
   train.pred.iter = train.pred.iter, train.prob.iter = train.prob.iter,
test.pred.iter = test.pred.iter, test.prob.iter = test.prob.iter,
specificity = specificity.value, auc = auc.value, average.bcr =
avg.bcr, chosen.feature = feature,
   coefficients = coef.m, coefis = coefis, lambda = lambda,
lambda.iterasi.min = lambda.iterasi.min, train.prediction =
train.pred, train.probability = train.prob, test.prediction =
test.pred, test.probability = test.prob, test.target=ttest,
last.prob=prob[,iter-1]))
}
```

Lampiran 33 Surat Keterangan Pengambilan Data

SURAT PERNYATAAN

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

Nama : Charles Rudiyanto
NRP : 062115 40000 0030

menyatakan bahwa data yang digunakan dalam Tugas Akhir/ ~~Thesis~~ ini merupakan data sekunder yang diambil dari penelitian / ~~buku/ Tugas Akhir/ Thesis/ Publikasi/ lainnya~~ yaitu:

Sumber : Data penelitian Ariyasu, *et al.*, (2014) dengan judul "*Design and Synthesis of 8-Hydroxyquinoline-Based Radioprotective Agents*"

Keterangan : Data proteksi radiasi dengan 84 observasi (senyawa), dua variabel respon yaitu tingkat kematian sel kanker dan kelas proteksi radiasi, serta 217 variabel prediktor yang merupakan penyusun senyawa.

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

Mengetahui,
Pembimbing Tugas Akhir

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

Surabaya, Juli 2019
Mahasiswa

Charles Rudiyanto
NRP. 062115 4000 0030

BIODATA PENULIS

Penulis dengan nama lengkap Charles Rudiyanto dilahirkan di Bekasi pada 22 November 1996. Penulis menempuh pendidikan formal di SDN Pulogebang 21 Pagi Jakarta, SMP Negeri 172 Jakarta, dan SMA Negeri 12 Jakarta. Kemudian penulis diterima sebagai Mahasiswa Departemen melalui jalur SNMPTN pada tahun 2015. Selama masa perkuliahan, penulis aktif di kepanitiaan sebagai Fasilitator Acara GERIGI ITS 2016, sie acara Natal dan Paskah PMK ITS 2016 dan sie acara Cerdas Bersama Statistika (CERITA) 2017. Penulis juga aktif dalam organisasi kemahasiswaan Himpunan Mahasiswa Statistika ITS sebagai staff Departemen Dalam Negeri (DAGRI) periode 2016/2017, organisasi fakultas FMKSD sebagai Kabiro Minat dan Bakat Departemen Internal periode 2017/2018. Dibidang akademik, penulis diberi kesempatan untuk menjadi semifinalis pada Pekan Analisis Statistika tahun 2019 yang diselenggarakan oleh Universitas Mulawarman. Apabila pembaca ingin memberi kritik dan saran serta diskusi lebih lanjut terkait Tugas Akhir ini, dapat menghubungi penulis melalui *e-mail* charlesrudiyanto@gmail.com atau melalui nomor 081382647510.