e-ISSN 2962-6129
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 Agustus 2022

Penerapan *Borderline*-SMOTE dan *Grid Search* pada *Bagging*-SVM untuk Klasifikasi Penyakit Diabetes

Trianto[1], Anita Muliawati[2], Helena Nurramdhani Irmanda[3],
Prodi S1 Informatika / Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Daerah Khusus Ibukota Jakarta 12450
[1]trianto@upnvj.ac.id, [2]anitamuliawati@upnvj.ac.id, [3]helenairmanda@upnvj.ac.id

Abstrak. Diabetes adalah penyakit berbahaya yang dapat menyebabkan kelumpuhan hingga mengancam jiwa penderitanya. Pada tahun 2019, Indonesia menempati peringkat ke-7 dari 10 negara dengan jumlah penderita terbanyak yaitu 10,7 juta penderita. Agar terhindar dari bahaya diabetes dapat menggunakan *data mining* untuk membangun model klasifikasi yang akan digunakan untuk melakukan prediksi dini. Algoritma klasifikasi yang bisa digunakan untuk membentuk model prediksi dini adalah *support vector machine* (SVM), sayangnya SVM memiliki kelemahan ketika diberikan data dengan kelas yang tidak seimbang dan sulitnya menentukan parameter optimal. Untuk mengatasi kelemahan tersebut akan digunakan algoritma *grid search*, *borderline*-SMOTE dan *bagging*. Hasil penelitian menunjukan bahwa model yang dibentuk dengan algoritma SVM, *bagging*, *borderline*-SMOTE dan *grid search* mendapat akurasi sebesar 92,1%, nilai *precision* sebesar 95,51% untuk kelas sehat dan 86,12% untuk kelas diabetes, nilai *recall* sebesar 92,32% untuk kelas sehat dan 91,66% untuk kelas diabetes, dan nilai *f1-score* sebesar 93,39% untuk kelas sehat dan 88,81% untuk kelas diabetes.

Kata Kunci: klasifikasi, diabetes, SVM, *bagging*, *borderline*-SMOTE, *grid search*

1 Pendahuluan

Diabetes adalah penyakit yang disebabkan oleh kadar gula darah yang tidak terkontrol dalam tubuh, yang mencegah pankreas memproduksi insulin yang cukup. Insulin sendiri merupakan hormon yang bertugas untuk membawa glukosa kepada sel-sel tubuh sebagai bahan bakar yang diperlukan oleh sel tersebut [8]. Pada tahun 2019, Indonesia menduduki peringkat ke-2 di Kawasan Pasifik Barat dan ke-7 diantara 10 negara paling terdampak dengan 10,7 juta orang terkena dampak dari 17,2 juta orang dewasa [2]. Semakin lama seseorang mengidap penyakit diabetes akan menyebabkan risiko terjadi komplikasi semakin tinggi. Komplikasi pada penyakit diabetes dapat menimbulkan penyakit kardiovaskular, stroke, jantung, kerusakan pada pembuluh darah, penglihatan, pendengaran, kulit, ginjal, kaki, dan dapat menyebabkan depresi [12]. Salah satu tindakan pencegahan untuk menghindari bahaya dari penyakit diabetes adalah dengan melakukan prediksi dini yang dapat memprediksi penyakit diabetes. Dengan mengetahui penyakit diabetes sejak dini dapat membantu menghindari dan mengobati penyakit yang disebabkan oleh diabetes. Metode yang dapat digunakan untuk melakukan prediksi apakah seseorang menderita diabetes adalah dengan memanfaatkan *data mining*. *Data mining* adalah metode pemrosesan data yang digunakan untuk menemukan informasi agar dapat menyelesaikan suatu masalah dengan melakukan analisis terhadap kumpulan data. Selain analisis, *data mining* juga berfungsi untuk mencari pola yang dapat memberikan informasi berdasarkan data yang diberikan [18]. Beberapa metode pemrosesan data yang biasa digunakan dalam *data mining* adalah regresi, klasifikasi, *clustering*, asosiasi dan masih banyak lagi metode lainnya [28]. Dari metode-metode pemrosesan data tersebut, metode yang digunakan untuk medeteksi penyakit diabetes adalah klasifikasi. Klasifikasi adalah metode yang digunakan untuk menemukan suatu pola yang mampu mendeskripsikan dan membedakan kelas-kelas dalam suatu kumpulan data [18]. Salah satu algoritma klasifikasi yang umum digunakan untuk mendeteksi diabetes adalah *support vector machine* [9].

Algoritma klasifikasi *Support vector machine* (SVM) adalah algoritma yang mampu menemukan *hyperplane* optimal yang dapat memisahkan setiap kelas pada data [5]. Namun, pengklasifikasian data dengan distribusi kelas yang tidak seimbang dapat menyebabkan SVM menghasilkan model klasifikasi yang buruk [1]. Kekurangan lain yang dimiliki oleh SVM adalah pemilihan *hyperparameter* optimal yang sulit [27]. Untuk dapat mengatasi kekurangan SVM saat melakukan klasifikasi pada data yang tidak seimbang maka digunakan *borderline*-SMOTE untuk melakukan *oversampling* pada data. Metode *oversampling* berfungsi untuk

meningkatkan jumlah data pada kelas minor agar distribusi kelas menjadi seimbang [14]. Sayangnya, metode *oversampling* memiliki kekurangan karena dapat menyebabkan *overfitting*. Hal ini terjadi karena data yang dihasilkan dari metode *oversampling* terlalu fokus pada *training data* dari *dataset* tertentu sehingga tidak dapat memprediksi dengan tepat apabila diberikan *dataset* lain yang serupa [21]. Untuk dapat mengatasi masalah optimasi *hyperparameter* SVM dan *overfitting* yang dihasilkan dari metode *oversampling* dapat diselesaikan dengan menggunakan *grid search*. *Grid search* akan mencari *hyperparameter* optimal dengan melakukan pencarian lengkap terhadap subset ruang *hyperparameter* yang telah ditentukan [23]. Kemudian untuk meningkatkan hasil klasifikasi SVM menjadi lebih baik dan menghindari *overfitting* serta mengurangi variansi maka akan digunakan algortitma *bagging*. *Bagging* bekerja dengan cara mengkombinasikan model klasifikasi dari *dataset training* yang telah di-*sampling* secara acak [15].

Penelitian sebelumnya yang dilakukan oleh Manurung pada tahun 2018 menerapkan metode hibrid dari *information gain* dan *bagging* untuk meningkatkan akurasi klasifikasi *support vector machine* pada skenario klasifikasi kelas biner yang menggunakan *dataset* diabetes. Metode yang digunakan dalam penelitian ini menghasilkan peningkatan akurasi klasifikasi dari 77,34% menjadi 82,14% [9]. Rousyati et al pada tahun 2021 juga melakukan penelitian pengaruh *bagging* terhadap performa algoritma klasifikasi *support vector machine* pada *dataset* diabetes. Hasil dari penelitian ini menunjukkan bahwa *bagging* dapat meningkatkan performa klasifikasi *support vector machine* khususnya akurasi, mulai dari 77,34% menjadi 77,47% [20]. Pada tahun 2020 Hairaini et al menerapkan algoritma k-*means*-SMOTE untuk menangani ketidakseimbangan kelas yang ada pada *dataset* diabetes, kemudian melakukan klasifikasi dengan algoritma *support vector machine*. Hasil kombinasi dari k-*means*-SMOTE dengan *support vector machine* mendapat akurasi sebesar 82% dan sensitivitas sebesar 77% [6]. Kemudian, Wahyu Nugraha pada tahun 2022 menggunakan *grid search* dan *cross validation* untuk mengoptimasi *hyperparameter* pada model prediksi agar dapat meningkatkan performa model. Dari hasil optimasi *hyperparameter* dengan menggunakan *dataset* diabetes, algoritma *support vector machine* mendapat *mean cross validation* sebesar 0,763 [26].

Berdasarkan uraian penelitian yang telah dijelaskan di atas, performa SVM yang dikombinasi dengan *bagging*, metode *oversampling data* dan *grid search* untuk mengatasi kekurangan yang dimiliki SVM ketika mebentuk model dari *dataset* diabetes memiliki performa yang baik. Dengan demikian, penelitian ini akan mencoba untuk meningkatkan performa klasifikasi *Support Vector Machine* dengan menerapkan *borderline*-SMOTE untuk mengatasi *dataset* yang tidak seimbang, *grid search* untuk mencari *hyperparameter* SVM yang optimal serta menghindari *overffiting*, dan *bagging* untuk menghindari *overfitting* serta mengurangi variansi.

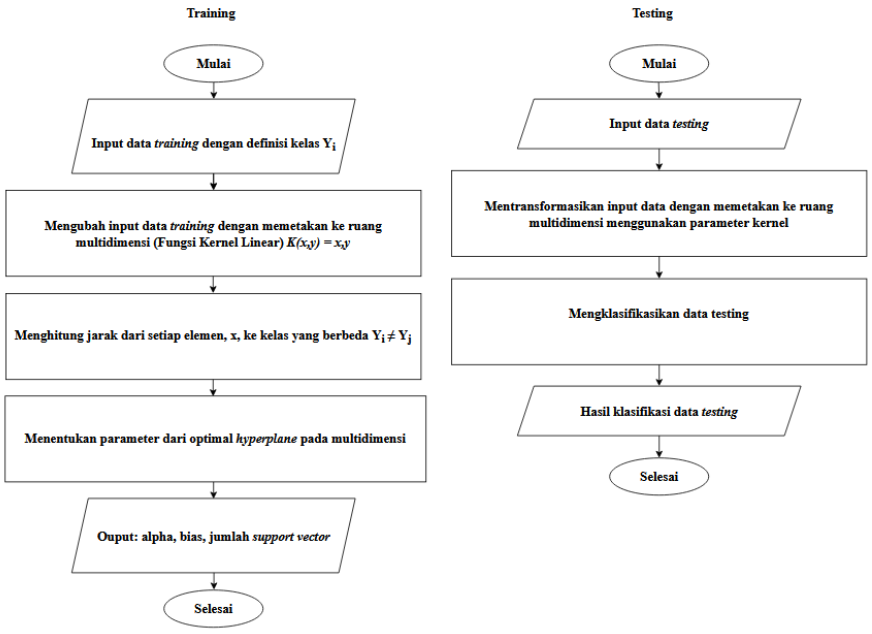
2 Tinjauan Pustaka

2.1 Diabetes

Diabetes Mellitus (DM) merupakan penyakit gangguan metabolisme yang terjadi karena produksi insulin di prankreas yang tidak dapat mencukupi kebutuhan tubuh atau insulin yang dihasilkan tidak mampu digunakan dengan baik oleh tubuh, sehingga kadar gula atau glukosa darah meningkat. Insulin adalah hormon yang bertugas memberi sinyal kepada sel tubuh untuk menyerap glukosa. Gejala yang biasa dikeluhkan oleh penderita DM antara lain *polifagia* (lapar berlebihan), *poliuria* (produksi urin berlebihan), *polydipsia* (haus berlebihan), penurunan berat badan, dan kesemutan. Penyebab penyakit DM adalah kekurangan insulin secara absolut atau relatif. Kekurangan insulin dalam tubuh dapat terjadi melalui 3 tahapan, yaitu pengaruh luar seperti bahan kimia atau virus yang merusak sel β di pankreas, reseptor glukosa pada kelenjar pankreas yang menurun dan reseptor insulin di dalam jaringan perifer yang rusak [4]. Menurut Nugroho dkk [15], terdapat 2 kategori utama yang ada pada penyakit diabetes. Diabetes Mellitus tipe 1 (DMT1) adalah tipe diabetes yang menyerang anak-anak hingga orang dewasa yang memiliki kadar insulin dalam peredaran darah berkurang akibat sel β yang menghasilkan insulin menghilang dari pankreas. Diabetes Mellitus tipe 2 (DMT2) terjadi karena tubuh tidak cukup menerima insulin, sehingga menyebabkan peningkatan konsentrasi glukosa. Sebanyak 90% dari keseluruhan penderita diabetes masuk kedalam DMT2.

2.2 *Support Vector Machine* (SVM)

Pada tahun 1992 di acara *Annual Workshop on Computational Learning Theory*, Vapnik bersama dengan Boser dan Guyon mempresentasikan algoritma yang mereka kembangkan bernama *Support Vector Machine* (SVM). SVM didasari pada kombinasi beberapa teori komputasi yang telah ada sebelumnya, salah satunya adalah *hyperplane* [9]. Dasar cara kerja dari SVM adalah mencari nilai maksimal dari batas *hyperplane*. Proses klasifikasi menggunakan SVM merupakan proses mencari nilai terbaik yang dapat memisahkan dua buah kelas data dari ruang input, nilai inilah yang disebut dengan *hyperplane*. Nilai *hyperplane* didapat dari hasil pengukuran *hyperplane* dengan margin yang merupakan jarak antara titik data dari perwakilan masing-masing kelas dengan *hyperplane*. Proses mencari perwakilan titik data yang menjadi margin untuk pembuatan *hyperplane* merupakan inti dari proses pelatihan SVM [16]. Proses klasfikasi pada algoritma SVM dibagi kedalam 2 tahapan yaitu tahap *training* dan tahap *testing* yang dapat dilihat pada Gambar 1.



Gambar 1. Tahap *Training* dan *Testing* Algoritma SVM [16]

Sayangnya, sangat jarang bagi permasalahan yang ada di dunia nyata memiliki data yang dapat terpisah secara linier. Untuk menyelesaikan permasalahan non-linier, SVM akan memetakan data kedalam ruang dimensi yang lebih tinggi dengan menggunakan fungsi kernel. Ada 4 fungsi kernel yang biasa digunakan SVM untuk menyelesaikan permasalahan *non-linear* maupun *linear* [4]. Keempat fungsi kernel tersebut adalah kernel *linear, polynomial, radial basis function* (RBF) dan *sigmoid*. Persamaan untuk tiap kernel tersebut adalah:

1. Kernel Sigmoid

$$K(x_i, x) = tanh(\gamma\, x_i^T x + r)^d, \gamma > 0 \quad \textbf{(1)}$$

2. Kernel Polynomial

$$K(x_i, x) = (\gamma\, x_i^T x + r)^d, \gamma > 0 \quad \textbf{(2)}$$

3. Kernel Linear

$$K(x_i, x) = x_i^T x \quad \textbf{(3)}$$

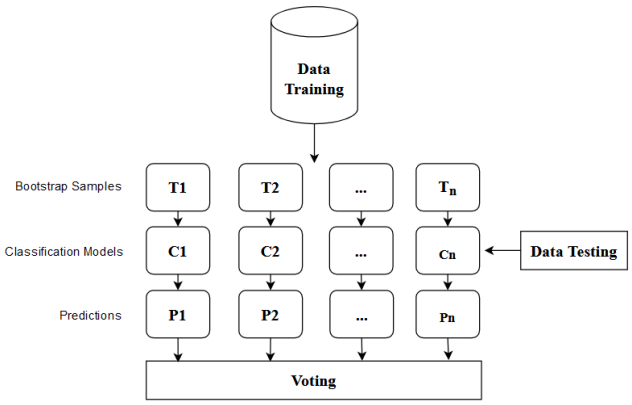
4. Kernel Radial Basis Function (RBF)

$$K(x_i, x) = \exp(-\gamma\, |x_i - x|^2), \gamma > 0 \quad \textbf{(4)}$$

Dimana dalam persamaan kernel di atas γ, r, dan d merupakan parameter kernel, serta parameter C sebagai penalti akibat kesalahan dalam klasifikasi untuk masing-masing kernel [10]. Performa klasifikasi dari algoritma SVM sangat bergantung dari nilai *hyperparameter* dan fungsi kernel yang digunakan. Nilai *hyperparameter* yang sangat berpengaruh dalam performa SVM adalah parameter *Cost* (C), besarnya nilai pada parameter ini dapat menghasilkan penalti yang besar terhadap klasifikasinya. Parameter selanjutnya adalah parameter *Gamma* (γ), parameter ini digunakan pada kernel RBF untuk mentrasformasi data *training* kedalam ruang fitur supaya nantinya dapat dioptimasi dengan metode *Lagrange Multipliers* sehingga menghasilkan nilai a yang dapat memperkirakan nilai koefisien w (bobot) atau b (bias) dan menentukan *support vector* pada model klasifikasi [7].

2.3 *Bagging*

Bagging atau *Bootstrap Aggregating* adalah metode yang bisa digunakan untuk membuat performa algoritma *machine learning* menjadi lebih baik [13]. *Bagging* diperkenalkan oleh Breiman pada tahun 1994 sebagai paduan dari teori *bootstrap* dan *aggregating* yang digabung menjadi satu [9]. *Bootstrap* akan membuat sub-*dataset* dengan melakukan *resampling* pada *dataset training*, kemudian *aggregating* dilakukan untuk mendapat nilai prediksi yang merupakan hasil gabungan beberapa nilai prediksi *sub-dataset* [22]. Nilai akhir prediksi didapat dengan melakukan *voting* melalui pengambilan suara terbanyak atau *bounded minority rule* [19].



Gambar 2. Tahapan *Bagging* [15]

2.4 *Borderline*-SMOTE

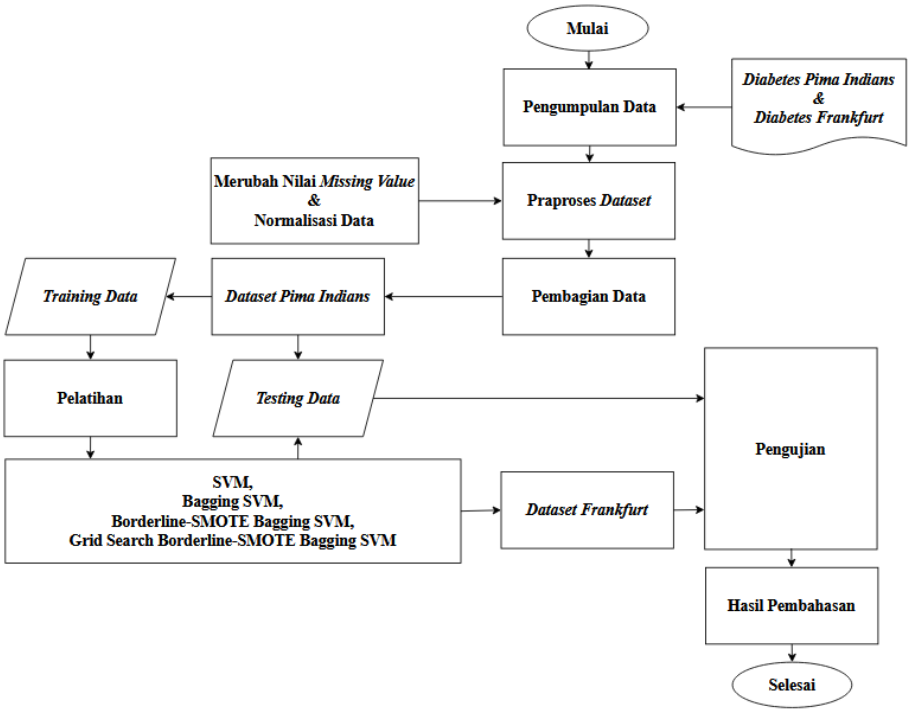
Borderline-SMOTE adalah metode pengembangan dari algoritma SMOTE yang akan melakukan *sampling* untuk mendapat sampel kelas minoritas yang berlebihan. Sebagai contoh *oversampling* kelas minoritas, k (yang diatur 5 pada SMOTE) tetangga yang paling dekat dari kelas yang sama dihitung, kemudian secara acak akan dipilih beberapa sampel sesuai dengan tingkat *oversampling*. Setelah itu akan dihasilkan *sampling* dari data sintetis baru antara sampel minoritas dengan tetangga terdekat yang terpilih. Hal ini berbeda dari metode pengambilan sampel yang dilakukan pada SMOTE, metode *borderline*-SMOTE hanya akan melakukan duplikasi kelas minoritas atau memperkuatnya [11]. Pada tahun 2009 Hien Nguyen, et al mengusulkan sebuah metode alternatif yang menggunakan SVM sebagai pengganti *k-nearest neighbors* untuk menemukan area *borderline*. Metode yang diusulkan Hien Nguyen, et al akan menghasilkan data sintetik kelas minor disepanjang *decision boundary* dikarenakan daerah tersebut sangat penting dalam memperkirakan *decision boundary* yang optimal. Jika data kelas minor yang terhitung masih kurang dari setengah data tetangga terdekatnya, maka akan dibuat data sintetik yang baru dengan tujuan untuk memperluas area dari kelas minor menuju kelas mayor [3]. Penelitian yang dilakukan Verdikha, et al [25] telah membuktikan bahwa *borderline*-SMOTE yang dikembangkan oleh Hien Nguyen, et al memberikan hasil kinerja peningkatan klasifikasi paling tinggi dibandingkan dengan algoritma SMOTE lainnya.

2.5 *Grid Search*

Grid Search merupakan algoritma pencarian menyuluruh terhadap subset ruang *hyperparameter* berdasarkan jumlah angka, nilai minimal (*lower bound*), dan nilai maksimal (*upper bound*) yang ada pada ruang subset [24]. Algoritma ini akan mencari nilai *hyperparameter* yang optimal dengan cara membagi jangkauan *hyperparameter* kedalam sebuah *grid* dan melalui seluruh titik kemungkinan yang ada. Dalam penerapannya, algoritma *grid search* akan bekerja sama dengan teknik *cross validation* pada data *training* sebagai metrik kinerja yang berguna untuk menghindari *classifier* menghasilkan model prediksi yang *overfitting*. Hal ini bertujuan untuk menentukan kombinasi dari tiap *hyperparameter* yang memberikan hasil optimal untuk *classifier* dan membuat *classifier* dapat melakukan prediksi pada data *testing* dengan akurat [23].

3 Metodologi

Model klasifikasi yang dibentuk dalam penelitian ini bertujuan untuk melakukan klasifikasi apakah pasien menderita diabetes atau tidak berdasarkan data-data yang dimiliki oleh pasien. Penelitian ini akan memakai 2 *dateset* diabetes yang berasal dari *Pima Indians* dan *Frankfurt*, dimana pembentukan model klasfikasi akan menggunakan *dataset Pima Indians* kedalam beberapa skenario. Beberapa skenario pembentukan model klasifikasi terjadi akan menggabungkan algoritma SVM, *bagging*, *borderline*-SMOTE dan *grid search*. Kemudian untuk melihat performa model klasifikasi terhadap data yang tidak digunakan saat proses pelatihan dan pengujian dengan *dataset Pima Indians*, maka akan dilakukan klasifikasi pada *dataset diabetes Frankfurt* menggunakan model klasifikasi yang telah dihasilkan sebelumnnya. Untuk gambaran sistem dapat dilihat pada Gambar 3.



Ganmbar 3. Gambaran umum Sistem Klasifikasi

Pengumpulan Data. Penelitian ini menggunakan data dari *dataset Pima Indians Diabetes* dan *dataset Frankfurt Diabetes* yang dapat diakses pada situs https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database dan https://www.kaggle.com/datasets/johndasilva/diabetes. *Dataset Pima Indians* akan dibagi menjadi *training data* yang akan digunakan untuk membentuk model klasifikasi dan *testing data* yang akan digunakan bersama dengan *dataset Frankfurt* untuk menguji performa model klasifikasi yang terbentuk dari *training data Pima Indians*. Atribut yang dimiliki kedua *dataset* ini merupakan diagnosis kondisi medis seseorang dan keterangan apakah memiliki penyakit diabetes atau tidak Atribut-atribut tersebut adalah adalah jumlah kehamilan pasien

wanita (preg), konsentrasi gula darah atau glukosa pasien selama 2 jam dalam tes toleransi glukosa (plas), tekanan darah diastolik (pres), seberapa tebal lapisan kulit yang berada pada trisep (skin), tingkat insulin yang dimiliki pasien dalam 2 jam penggunaan serum insulin (insu), indeks masa tubuh (mass), penderita penyakit diabetes dalam keluarga pasien (pedi), usia pasien (age) dan apakah pasien memiliki penyakit diabetes atau tidak (outcome). Berikut adalah perbandingan dari *dataset Pima Indians* dan *Frankfurt*.

Tabel 1. Perbandingan *Dataset Pima Indians* dan *Frankfurt*

Dataset	**Sehat**	**Diabetes**	**Total**
Pima Indians	500	268	768
Frankfurt	1316	684	200

Praproses *Dataset*. Dalam proses ini akan dilakukan praproses kepada kedua *dataset* yang digunakan. Praposes bertujuan untuk menemukan dan mengatasi data-data yang dianggap sebagai *missing value* karena data tersebut dapat menyebabkan masalah pada proses pelatihan. *Missing value* ditandai dengan nilai min 0 pada atribut yang seharusnya tidak boleh memiliki nilai 0. Atribut tersebut adalah *Glucose, BloodPressure, SkinThickness, Insulin* dan BMI. Nilai 0 pada atribut tersebut akan diganti dengan nilai *NaN* dan kemudian angka diganti berdasarkan nilai rata-rata distribusi yang dimiliki oleh atribut tersebut dengan *median* atau *mean* distribusinya. Untuk atribut *Glucose* dan *BloodPressure* akan digantikan oleh *mean* distribusinya sedangkan atribut lainnya akan digantikan oleh *median* distribusinya.

Setelah *missing value* teratasi, praproses selanjutnya adalah normalisasi data dengan mengubah data kedalam rentang nilai yang lebih kecil. Normalisasi data akan dilakukan dengan memanfaatkan *standard scaler* yang telah disediakan oleh *library sklearn*. *Standard scaler* akan mencegah adanya data dengan nilai yang terlalu besar dibandingkan nilai lainnya, karena perbedaan nilai tersebut dapat mengakibatkan proses *training* tidak berjalan sesuai keinginan [17]. Rumus perhitungan yang digunakan oleh standard scaler dapat dilihat pada persamaan 5, dimana $\bar{X}$ adalah rata-rata dari nilai sampel dan σ adalah standar deviasi.

$$Z = \frac{X_i - \bar{X}}{\sigma} \quad \textbf{(5)}$$

Pembagian Data. Dalam penelitian ini, akan ada 2 pembagian data dari *dataset Pima Indians* yaitu, *dataset Pima Indians* sebelum dilakukan *oversampling* yang memiliki 768 data dengan 500 data sehat (kelas mayor) dan 268 data diabetes (kelas minor) dan *dataset Pima Indians* yang telah dilakukan *oversampling* dengan *borderline*-SMOTE yang memiliki 500 data sehat dan 500 data diabetes. Proses pembagian data kedalam *training data* dan *testing data* akan dilakukan pada *dataset Pima Indians* dengan perbandingan 75% dari total untuk *training data* dan 25% sisanya untuk *testing data* yang dibagi secara *random* terhadap keseluruhan data. Pembagian ini berlaku untuk *dataset* yang tidak di-oversampling dan *dataset* yang telah di-*oversampling* dengan *borderline*-SMOTE. Setelah pembagian data kedalam *training data* dan *testing data*, *training data* akan digunakan untuk membuat model klasifikasi sedangkan *testing data* digunakan pada tahap klasifikasi untuk menguji performa dari model klasifikasi.

Pelatihan. Pada tahap ini akan dilakukan pembentukan model klasifikasi berdasarkan *training data* dari *dataset Pima Indians* yang sudah dilakukan praproses, dimana model klasifikasi yang dibentuk dalam tahapan ini akan menentukan tingkat keberhasilan penelitian. Pembentukan model klasifikasi yang dilakukan akan dibagi menjadi 4 skenario yang dapat dilihat pada tabel 2.

Tabel 2. Skenario Pembentukan Model

Skenario	Algortima	Keterangan
1	SVM	Pada skenario ini akan membentuk model klasifikasi berdasarkan *training data Pima Indians* dengan menggunakan algoritma SVM.
2	SVM, *Bagging*	Pada skenario ini akan menggabungkan algoritma SVM dan *bagging* untuk membentuk model klasifikasi berdasarkan *training data Pima Indians*.
3	SVM, *Bagging*, *Borderline*-SMOTE	Pada skenario ini akan melakukan *oversampling* terhadap *training data Pima Indians* dengan *borderline*-SMOTE sehingga jumlah *class data* menjadi seimbang, setelah itu data akan digunakan untuk membentuk model klasifikasi dengan algoritma SVM dan *bagging*.
4	SVM, *Bagging*, *Borderline*-SMOTE, *Grid Search*	Pada skenario ini, *grid search* akan digunakan untuk mengoptimasi *hyperparameter* yang dimiliki oleh algoritma SVM dan *bagging* pada saat membentuk model klasifikasi berdasarkan *training data Pima Indians* yang telah di-*oversampling* dengan *borderline*-SMOTE.

Pengujian. Untuk dapat melihat bagaimana performa yang dimiliki oleh tiap skenario model klasifikasi dapat menggunakan *confusion matrix* yang telah disediakan oleh *library sklearn.metrics.confusion_matrix*. *Confusion matrix* dapat digunakan untuk menampilkan informasi perbandingan antara hasil klasifikasi dengan data sebenarnya. Dari hasil *confusion matrix* dapat diketahui seberapa besar nilai akurasi, *precision*, *recall* dan *f1-score* yang dimiliki oleh tiap skenario model klasifikasi. Perhitungan nilai akurasi, *precision*, *recall* dan *f1-score* dapat dilakukan dengan menggunakan perhitungan yang telah disediakan oleh *library sklearn.metrics.accuracy_score* untuk akurasi, *sklearn.metrics.precision_score* untuk *precision*, *sklearn.metrics.recall_score* untuk *recall* dan *sklearn.metrics.f1_score* untuk *f1-score atau* menggunakan persamaan di bawah ini.

$$Akurasi = \frac{TP+TN}{TP+FN+FP+TN} \quad \textbf{(6)}$$

$$Precision = \frac{TP}{TP+FP} \quad \textbf{(7)}$$

$$Recall = \frac{TP}{TP+FN} \quad \textbf{(8)}$$

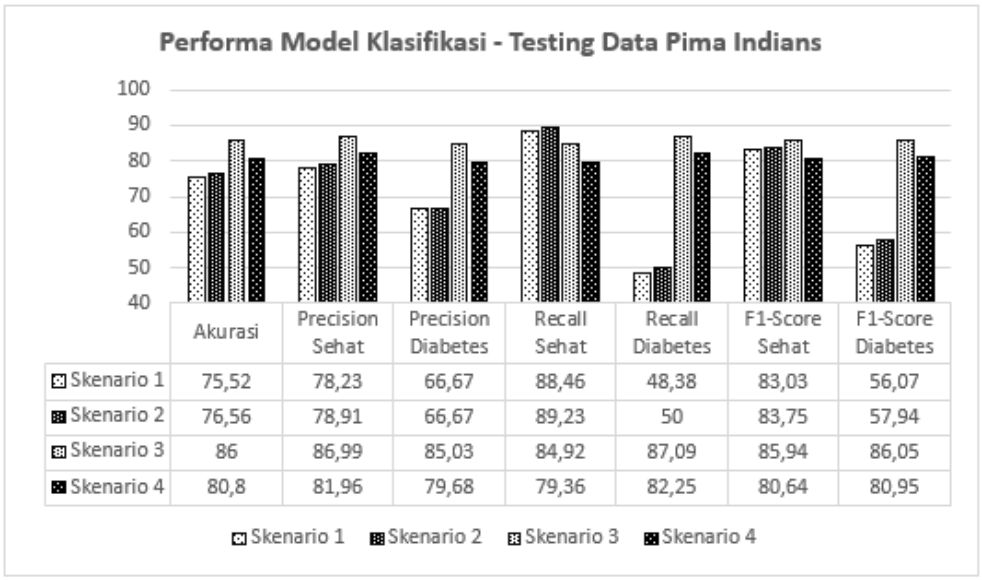
$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad \textbf{(9)}$$

Nilai-nilai ini akan dijadikan sebagai tolak ukur apakah model klasifikasi memiliki performa yang bagus atau tidak dalam melakukan klasifikasi penyakit diabetes.

Hasil Pembahasan. Dalam tahapan ini, penulis membahas hasil dari penelitian yang telah dilakukan. Pembahasan tersebut akan menjelaskan apakah model klasifikasi penyakit diabetes yang terbentuk sudah memiliki performa yang baik atau belum dengan melihat nilai akurasi, *precision*, *recall* dan *f1-score* yang dihitung dengan persamaan (6), (7), (8) dan (9). Kemudian menampilkan dan memberikan analisa dari masing-masing model klasifikasi penyakit diabetes.

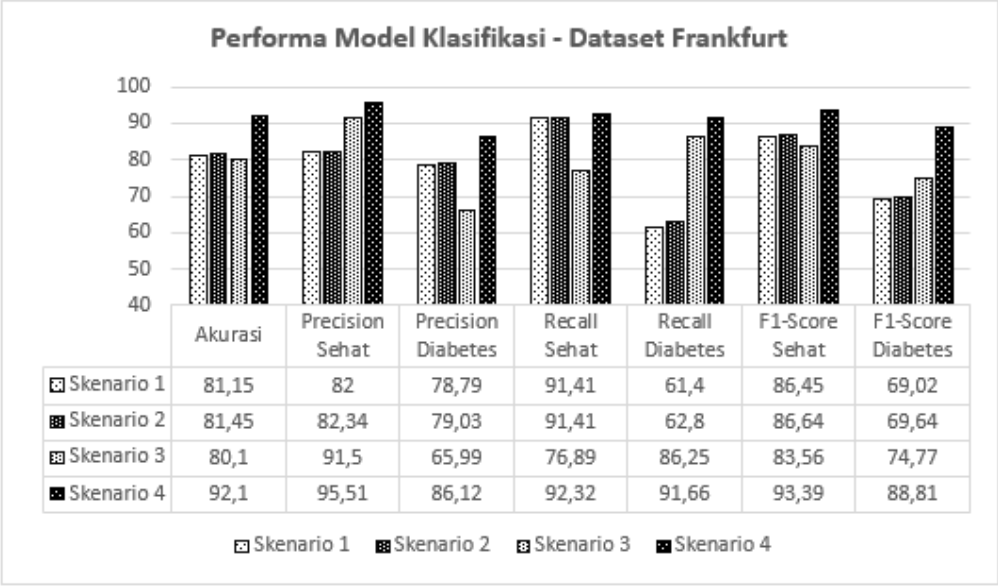
4 Hasil dan Pembahasan

Pada percobaan yang dilakukan dalam penelitian terdapat 4 skenario pembentukan model klasifikasi yaitu model SVM, model SVM dengan *bagging*, model SVM dan *bagging* dengan data yang telah di-oversampling oleh *borderline*-SMOTE dan model SVM dengan *bagging* dan *borderline*-SMOTE yang telah dioptimasi oleh *grid search* terhadap *training data Pima Indians*. Tiap skenario tersebut akan menggunakan data yang telah melewati tahapan praproses untuk menghilangkan *missing value* dan melakukan normalisasi data dengan *standard scaler*, tahapan pembagian data menjadi *training data* dan *testing data* untuk *dataset* yang tidak di-*oversampling* dan *dataset* yang telah di-*oversampling* dengan *borderline*-SMOTE, tahapan pelatihan pembentukan model klasifikasi menggunakan *training data Pima Indians* sesuai dengan skenario pada tabel 2., dan kemudian tiap skenario akan diuji untuk melihat bagaimana performanya dengan mengunakan *testing data Pima Indians* dan *dataset Frankfurt*. Performa dari tiap skenario dapat dilihat pada gambar 4. untuk performa model klasifikasi terhadap *testing Data Pima Indians* dan gambar 5. untuk performa model klasifikasi terhadap *Dataset Frankfurt*.



	Akurasi	Precision Sehat	Precision Diabetes	Recall Sehat	Recall Diabetes	F1-Score Sehat	F1-Score Diabetes
Skenario 1	75,52	78,23	66,67	88,46	48,38	83,03	56,07
Skenario 2	76,56	78,91	66,67	89,23	50	83,75	57,94
Skenario 3	86	86,99	85,03	84,92	87,09	85,94	86,05
Skenario 4	80,8	81,96	79,68	79,36	82,25	80,64	80,95

Ganmbar 4. Performa Model Klasifikasi Terhadap *Testing Data Pima Indians*



	Akurasi	Precision Sehat	Precision Diabetes	Recall Sehat	Recall Diabetes	F1-Score Sehat	F1-Score Diabetes
Skenario 1	81,15	82	78,79	91,41	61,4	86,45	69,02
Skenario 2	81,45	82,34	79,03	91,41	62,8	86,64	69,64
Skenario 3	80,1	91,5	65,99	76,89	86,25	83,56	74,77
Skenario 4	92,1	95,51	86,12	92,32	91,66	93,39	88,81

Ganmbar 5. Performa Model Klasifikasi Terhadap *Dataset Frankfurt*

Skenario 1. Pada skenario 1 model klasifikasi yang terbentuk adalah model SVM dengan menggunakan *training data Pima Indians* yang belum dilakukan *oversampling* oleh *borderline*-SMOTE. *Hyperparameter* yang digunakan adalah *default hyperparameter* yang telah ditentukan oleh *library sklearn.svm.SVC* pada *versi scikit-learn 1.1.1*. Dikarenakan model ini dilatih dengan menggunakan *training data Pima Indians* yang memiliki ketidakseimbangan antara kelas Sehat dan Diabetes yang menyebabkan model ini tidak dapat melakukan prediksi terhadap kelas Diabetes dengan baik. Hal ini terjadi karena model skenario 1 memiliki nilai *precision*, *recall*, dan *f1-score* kelas Diabetes yang rendah ketika melakukan klasifikasi *testing data Pima Indians*, dengan nilai *precision* kelas Diabetes sebesar 66,67%, nilai *recall* kelas Diabates sebesar 48,38% dan nilai *f1-score* kelas Diabetes sebesar 56,07% yang dapat dilihat pada gambar 4. Pada klasifikasi *dataset Frankfurt* bersadasarkan gambar 5., model ini juga mendapat nilai *recall* kelas Diabetes yang rendah, dengan nilai 61,4%. Dari hasil tersebut membuktikan bahwa model skenario 1 tidak dapat melakukan prediksi terhadap kelas Diabetes dengan baik.

Skenario 2. Pada Skenario 2 model klasifikasi yang terbentuk adalah model *bagging*-SVM dengan menggunakan menggunakan *training data Pima Indians* yang belum dilakukan *oversampling* oleh *borderline*-SMOTE. *Hyperparameter* untuk algoritma SVM akan sama seperti yang digunakan pada skenario 1 dan *algoritma bagging* akan menggunakan *default hyperparameter* yang telah ditentukan oleh *library sklearn.ensemble.BaggingClassifier* pada *versi scikit-learn 1.1.1*, kecuali 1 *hyperparameter* algoritma *bagging* yaitu *n_estimators=100* yang menyatakan bahwa akan ada 100 *bag* dalam pembentukan model *bagging*-SVM. Dengan menerapkan *bagging*, performa model skenario ini mengalami peningkatan daripada performa model skenario sebelumnya, walaupun hanya sedikit. Tetapi sama seperti model skenario 1, model skenario ini juga dilatih dengan menggunakan *training data Pima Indians* yang memiliki ketidakseimbangan antara kelas Sehat dan Diabetes yang menyebabkan model ini tidak dapat melakukan prediksi terhadap kelas Diabetes dengan baik. Hal ini terjadi karena model skenario 2 memiliki nilai *precision* kelas Diabetes sebesar 66,67%, nilai *recall* kelas Diabates sebesar 50,0% dan nilai *f1-score* kelas Diabetes sebesar 57,94% pada saat melakukan klasifikasi *testing data Pima Indians* yang dapat dilihat pada gambar 4. Pada klasifikasi *dataset frankfurt* bersadasarkan gambar 5., model ini juga mendapat nilai *recall* kelas Diabetes yang rendah, dengan nilai 62,80%. Sama seperti model skenario 1, model skenario 2 tidak dapat melakukan prediksi terhadap kelas Diabetes dengan baik. Walaupun begitu, model skenario ini masih lebih baik daripada model skenario 1.

Skenario 3. Pada skenario 3 model klasifikasi yang terbentuk adalah model *bagging*-SVM yang sama seperti skenario 2 dengan data *borderline*-SMOTE. Model dalam skenario ini merupakan model *bagging*-SVM yang dilatih dengan *training data Pima Indians* yang telah di-*oversamping* oleh *borderline*-SMOTE yang diusulkan oleh Hien Nguyen, et al sehingga kelas Sehat dan kelas Diabetes memiliki jumlah yang sama. Algoritma *borderline*-SMOTE usulan Hien Nguyen, et al dapat digunakan dengan meng-*import SVMSMOTE* dari *library imblearn.over_sampling*. *Hyperparameter* yang digunakan adalah *default hyperparameter* yang telah ditentukan oleh library imblearn.over_sampling.SVMSMOTE versi 0.9.1.

Karena masalah ketidakseimbangan data telah teratasi, model dalam skenario ini dapat melakukan prediksi pada kelas Sehat maupun Diabetes dengan baik. Hal ini terjadi karena model skenario 3 memiliki nilai akurasi sebesar 86%, nilai *precision* kelas Sehat sebesar 86,99% dan kelas Diabetes sebesar 85,03%, nilai *recall* kelas Sehat sebesar 84,92% dan kelas Diabetes sebesar 87,09% kemudian nilai *f1-score* kelas Sehat sebesar 85,94% pada saat melakukan klasifikasi *testing data Pima Indians* yang dapat dilihat pada gambar 4. Tetapi Sayangnya, model skenario 3 tidak dapat mengeneralisasi dengan baik karena mengalami *overfitting*. Bersadasarkan gambar 5., indikasi *overfitting* dapat terlihat jelas ketika melakukan klasifikasi terhadap *dataset Frankfurt* dengan melihat nilai *precision* kelas Diabetes. Masalah *overfitting* terjadi karena model skenario 3 yang dibuat terlalu fokus pada data sintetis yang dihasilkan oleh *oversampling* dengan *borderline*-SMOTE yang dimiliki oleh *training data Pima Indians*, hingga model tidak dapat melakukan klasifikasi dengan tepat pada *dataset Frankfurt*.

Skenario 4. Pada skenario 4 model klasifikasi yang terbentuk adalah model *bagging*-SVM dengan data *borderlin*e-SMOTE yang sama seperti skenario 3 akan dioptimasi *hyperparameter*-nya dengan algoritma *grid search* yang telah disediakan oleh *library sklearn.model_selection.GridSearchCV* yang ada pada *scikit-learn versi 1.1.1*. *Grid search* akan mengoptimasi *hyperparameter* untuk *algoritma bagging* adalah *n_estimators* yang menyatakan banyaknya *bag*, sedangkan *hyperparameter* pada algoritma SVM adalah *kernel* untuk memetakan

data dalam ruang dimensi yang lebih tinggi, C yang berkaitan dengan *margin* pada SVM dimana besarnya nilai C maka *margin* dalam model SVM akan semakin kecil, dan *gamma* yang mempengaruhi model karena besarnya nilai *gamma* dapat mengakibatkan *overfitting* atau nilai gamma yang kecil dapat menghasilkan model yang tidak dapat mencakup kompleksitas data. *Grid search* akan mencari model dengan performa terbaik dari kombinasi hyperparameter yang diberikan. Dalam penelitian ini, nilai yang digunakan untuk tiap hyperparameter dapat dilihat pada tabel 3.

Tabel 3. *Hyperparameter* Algoritma *Bagging* dan SVM

Algoritma	***Hyperparameter***	
Bagging	*n_estimators*	10, 50, 100
SVM	kernel	*linear, rbf, poly, sigmoid*
	C	0.01, 0.1, 1.0, 10.0, 100.0
	gamma	1.0, 0.1, 0.01, 0.001

Dari tabel diatas, banyaknya kombinasi *hyperparameter* yang harus ditelusuri oleh *grid search* adalah sebanyak 240 kombinasi. Banyaknya kombinasi tersebut juga akan dikali dengan *cross validation* yang diperlukan oleh *grid search*, yaitu sebanyak 5 kali dan membuat kombinasi *hyperparameter* yang harus ditelusuri menjadi 1200 kombinasi. Setelah proses pelatihan selesai, maka *grid search* akan memberikan informasi *hyperparameter* mana yang memberikan hasil paling baik. Berikut *hyperparameter* terbaik untuk algoritma SVM dan *bagging* yang digunakan untuk membentuk model klasifikasi pada skenario 4.

Tabel 4. *Hyperparameter* terbaik untuk algoritma SVM dan *Bagging*

Algoritma	***Hyperparameter***	
Bagging	*n_estimators*	50
SVM	kernel	*rbf*
	C	1.0
	gamma	1.0

Dengan memanfaatkan *grid search*, model skenario 4 menjadi model klasifikasi dengan performa yang paling baik daripada model skenario lain. Berdasarkan gambar 4., walaupun tidak lebih baik dari model skenario 3, model skenario 4 juga dapat melakukan klasifikasi yang efektif terhadap terhadap *testing data Pima Indians*. Model ini juga memiliki performa yang paling baik pada *dataset Frankfurt* yang dapat dilihat pada gambar 5. Performa model skenario 4 yang baik terjadi karena pada model skenario ini telah mengatasi permasalahan yang dimiliki oleh model-model pada skanerio sebelumnya. Dengan demikian, model ini dapat melakukan prediksi pada kelas Sehat maupun Diabetes terhadap *testing data Pima Indians* maupun *dataset Frankfurt* dengan baik.

5 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dengan menerapkan algoritma SVM, *bagging*, *borderline*-SMOTE dan *grid search* untuk melakukan klasifikasi penyakit diabetes, maka dapat diambil kesimpulan sebagai berikut:

1. Penerapan algortima SVM, *bagging*, *borderline*-SMOTE dan *grid search* untuk membentuk model klasifikasi penyakit diabetes dengan menggunakan *training data* yang berasal dari *dataset Pima Indians*. Skenario pembentukan model yang terjadi terdiri dari model *support vector machine*, *model*

support vector machine dengan *bagging*, model *support vector machine* dan *bagging* dengan *training data* yang telah di-*oversampling* oleh *borderline*-SMOTE dan model *support vector machine* dengan *bagging* dan *borderline*-SMOTE yang telah dioptimasi oleh *grid search*. Kemudian model dari tiap skenario diuji dengan menggunakan *testing data* yang berasal dari *dataset Pima Indians* dan *dataset Frankfurt*.

2. Performa pengujian model pada skenario 4 yang terbentuk dari kolaborasi antara algoritma SVM, *bagging*, *borderline*-SMOTE dan *grid search* merupakan model klasifikasi yang memiliki generalisasi paling baik daripada model pada skenario lainnya. Hasil dari evaluasi terhadap *testing data* mendapat nilai akurasi sebesar 80,8%, nilai *precision* sebesar 81,96% untuk kelas Sehat dan 79,68% untuk kelas Diabetes, nilai *recall* sebesar 79,36% untuk kelas Sehat dan 82,25% untuk kelas Diabetes, dan nilai *f1-score* sebesar 80,64% untuk kelas Sehat dan 80,95% untuk kelas Diabetes. Hasil tersebut memang lebih rendah daripada yang didapat oleh model skenario 3, tetapi model skenario 3 mengalami penurunan performa pada saat melakukan klasifikasi *dataset Frankfurt*. Sedangkan hasil evaluasi model skenario 4 terhadap *dataset Frankfurt* mengalami kenaikan daripada hasil evaluasi terhadap *testing data*. Hasil evaluasi model skenario 4 terhadap *dataset Frankfurt* mendapat akurasi sebesar 92,1%, nilai *precision* sebesar 95,51% untuk kelas Sehat dan 86,12% untuk kelas Diabetes, nilai *recall* sebesar 92,32% untuk kelas Sehat dan 91,66% untuk kelas Diabetes, dan nilai *f1-score* sebesar 93,39% untuk kelas Sehat dan 88,81% untuk kelas Diabetes.

6 Daftar Pustaka

[1] Amelia, O. D., Soleh, A. M., & Rahardiantoro, S. (2018). Pemodelan Support Vector Machine Data Tidak Seimbang Keberhasilan Studi Mahasiswa Magister IPB. *Xplore: Journal of Statistics*, *2*(1), 33–40. https://doi.org/10.29244/xplore.v2i1.76

[2] Atlas, I. D. F. D. (2019). International Diabetes Federation. In *The Lancet* (Vol. 266, Issue 6881). https://doi.org/10.1016/S0140-6736(55)92135-8

[3] Brownlee, Jason (2020, Maret 17). Machinelearningmastery. Dipetik Mei 22, 2022, dari https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification.

[4] Fathur Rahman, I. (2020). *Implementasi Metode Svm, Mlp Dan Xgboost Pada Data Ekspresi Gen*. https://dspace.uii.ac.id/handle/123456789/23679

[5] Gazni, H. A. Al. (2020). *Optimasi Algoritma Support Vector Machine Berbasis Algoritma K-Means Dan Particle Swarm Optimization Pada Diagnosis Penyakit Ginjal Kronis*.

[6] Hairani, H., Saputro, K. E., & Fadli, S. (2020). K-means-SMOTE for handling class imbalance in the classification of diabetes with C4.5, SVM, and naive Bayes. *Jurnal Teknologi Dan Sistem Komputer*, *8*(2), 89–93. https://doi.org/10.14710/jtsiskom.8.2.2020.89-93

[7] Handayani, A., Jamal, A., & Septiandri, A. A. (2017). *Evaluasi Tiga Jenis Algoritme Berbasis Pembelajaran Mesin untuk Klasifikasi Jenis Tumor Payudara*. *6*(4), 394–403.

[8] Howsalya Devi, R. D., Bai, A., & Nagarajan, N. (2020). A novel hybrid approach for diagnosing diabetes mellitus using farthest first and support vector machine algorithms. *Obesity Medicine*, *17*, 100152. https://doi.org/10.1016/j.obmed.2019.100152

[9] Manurung, I. H. G. (2018). Hibrid Metode Information Gaindan Bagging Dalam Klasifikasi Data Menggunakan Support Vector Machine. In *Analisis Kesadahan Total dan Alkalinitas pada Air Bersih Sumur Bor dengan Metode Titrimetri di PT Sucofindo Daerah Provinsi Sumatera Utara* (Vol. 2).

[10] Manurung, J. (2018). *Optimasi Parameter pada Support Vector Machine dengan Algoritma Genetika untuk Penilaian Risiko Kredit*.

[11] Marjones H H Sihombing. (2019). *KOMBINASI ALGO RITM A k-NN DAN SMOTE DALAM KLASIFIK ASI DATA TIDAK SEIMBANG*.

[12] Mayo Clinic. (2018). Diabetes. Dipetik November 21, 2021, dari https://www.mayoclinic.org/diseases-conditions/diabetes/symptoms-causes/syc-20371444

[13] Moukhafi, M., El Yassini, K., & Bri, S. (2018). Mining network traffics for intrusion detection based on Bagging ensemble Multilayer perceptron with Genetic algorithm optimization. *International Journal of Computer Science and Network Security*, *18*(5), 59–66. http://search.ijcsns.org/02_search/02_search_03.php?number=201805009

[14] Muthahari, W. (2018). *Analisis teknik resampling menggunakan synthetic minority oversampling technique (smote) untuk melatih support vector machine (svm) wadudi muthahari*

[15] Nugroho, A., & Religia, Y. (2021). Analisis Optimasi Algoritma Klasifikasi Naive Bayes menggunakan Genetic Algorithm dan Bagging. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(3), 504–510. https://doi.org/10.29207/resti.v5i3.3067

[16] Prasetyo, E. (2013). Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab. Yogyakarta : Andi.

[17] Prasetyo, V. R., Mercifia, M., Averina, A., Lauren, Sunyoto, & Budiarjo. (2022). Prediksi Rating Film Pada Website IMDB Menggunakan Metode Neural Network. *Jurnal Ilmiah NERO*, *7*(1), 1–8
[18] Pristyanto, Y. (2019). Penerapan Metode Ensemble Untuk Meningkatkan Kinerja Algoritme Klasifikasi Pada Imbalanced Dataset. *Jurnal Teknoinfo*, *13*(1), 11. https://doi.org/10.33365/jti.v13i1.184.
[19] Riyanto, U. (2019). Analisis Perbandingan Algoritma Naive Bayes Dan Support Vector Machine Dalam Mengklasifikasikan Jumlah Pembaca Artikel Online. *JIKA (Jurnal Informatika)*, *2*(2), 62–72. https://doi.org/10.31000/.v2i2.1521
[20] Rousyati, R., Rais, A. N., Rahmawati, E., & Amir, R. F. (2021). Prediksi Pima Indians Diabetes Database Dengan Ensemble Adaboost Dan Bagging. *EVOLUSI : Jurnal Sains Dan Manajemen*, *9*(2), 36–42. https://doi.org/10.31294/evolusi.v9i2.11159
[21] Saputra, P. Y., Abdullah, M. Z., & Kirana, A. P. (2021). Improvisasi Teknik Oversampling MWMOTE Untuk Penanganan Data Tidak Seimbang. *Jurnal Media Informatika Budidarma*, *5*(2), 398. https://doi.org/10.30865/mib.v5i2.2811
[22] Siringoringo, R., & Kelana Jaya, I. (2018). *Ensemble Learning Dengan Metode Smote Bagging Pada Klasifikasi Data Tidak Seimbang*. *3*(2), 75–81.
[23] Sulistiana. (2020). *Optimasi Support Vector Machine (SVM) Menggunakan Grid Search dan Unigram Guna Mengingkatkan Akurasi Review Pada Situs E-Commerce*.
[24] Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *14*(4), 1502. https://doi.org/10.12928/telkomnika.v14i4.3956
[25] Verdikha, N. A., Adji, T. B., & Permanasari, A. E. (2018). Komparasi Metode Oversampling Untuk Klasifikasi Teks Ujaran Kebencian. *Seminar Nasional Teknologi Informasi Dan Multimedia 2018*, 85–90.
[26] Wahyu Nugraha, A. S. (2022). Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search. *SISTEMASI: Jurnal Sistem Informasi*, *11*, 391–401.
[27] Wahyuni, D. (2019). *Optimasi parameter support vector machine (svm) classifier menggunakan firefly algorithm (ffa) optimization untuk klasifikasi mri tumor otak*.
[28] Yunial, A. H. (2020). Analisis Optimasi Algoritma Klasifikasi Support Vector Machine, Decision Trees, dan Neural Network Menggunakan Adaboost dan Bagging. *Jurnal Informatika Universitas Pamulang*, *5*(3), 247. https://doi.org/10.32493/informatika.v5i3.6609