Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2023107234
Vol. 10, No. 6, Desember 2023, hlm. 1273-1284 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

PENYEIMBANGAN KELAS SMOTE DAN SELEKSI FITUR *ENSEMBLE* FILTER PADA *SUPPORT VECTOR MACHINE* UNTUK KLASIFIKASI PENYAKIT *LIVER*

Muhammad Amir Nugraha[1], Muhammad Itqan Mazdadi[*2], Andi Farmadi[3], Muliadi[4], Triando Hamonangan Saragih[5]

[1,2,3,4,5]Universitas Lambung Mangkurat, Banjarmasin
Email: [1]muhammadamirngrh@gmail.com, [2]mazdadi@ulm.ac.id, [3]andifarmadi@gmail.com, [4]muliadi@ulm.ac.id, [5]triando.saragih@ulm.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 05 Mei 2023, diterima untuk diterbitkan: 28 November 2023)

Abstrak

Liver merupakan salah satu organ penting dalam tubuh manusia yang berperan dalam proses metabolisme tubuh. Mengutip artikel dari situs *American Liver Foundation*, pada tahun 2020 sebanyak 51.642 orang dewasa di Amerika Serikat meninggal akibat penyakit *liver*. Data hasil tes fungsi *liver* dari laboratorium dapat digunakan untuk mendiagnosis penyakit *liver*. Klasifikasi penyakit *liver* pada pasien perlu dilakukan dengan baik karena hasilnya dapat membantu dalam diagnosis awal apakah seorang pasien mengidap penyakit *liver*. Berdasarkan penelitian sebelumnya, metode *Support Vector Machine* (SVM) paling baik dalam mengklasifikasikan pasien penyakit *liver*. Namun, SVM memiliki kelemahan ketika diterapkan pada *dataset* dengan kelas yang tidak seimbang dan tidak bekerja secara akurat ketika terlalu banyak fitur yang tidak relevan digunakan. Untuk menyeimbangkan kelas pada *dataset*, digunakan metode *Synthetic Minority Oversampling Technique* (SMOTE). Sedangkan untuk seleksi fitur dilakukan menggunakan metode *Ensemble* Filter, terdiri dari metode *Information Gain*, *Gain Ratio*, dan *Relief-F* untuk menangani fitur-fitur tidak relevan. Berdasarkan hasil pengujian, penerapan SMOTE dan *Ensemble* Filter pada metode klasifikasi SVM memberikan hasil terbaik dengan nilai *accuracy* sebesar 85% dan AUC sebesar 0,850. Pengujian tersebut dapat membuktikan jika SMOTE pada penyeimbangan kelas dan *Ensemble* Filter pada seleksi fitur dapat meningkatkan performa klasifikasi dari metode SVM.

Kata kunci: *Liver,* Klasifikasi, SVM, SMOTE, *Ensemble* Filter

SMOTE CLASS BALANCING AND ENSEMBLE FILTER FEATURE SELECTION IN SUPPORT VECTOR MACHINE FOR LIVER DISEASE CLASSIFICATION

Abstract

The liver is one of the important organs in the human body that plays a role in the body's metabolic processes. Quoting an article from the American Liver Foundation website, in 2020, as many as 51.642 adults in the United States died from liver disease. Liver function test data from the laboratory can be used to diagnose liver disease. Classification of liver disease in patients needs to be done well because the results can help in the initial diagnosis of whether a patient has liver disease. Based on previous research, the Support Vector Machine (SVM) method best classifies liver disease patients. However, SVM has weaknesses when applied to datasets with unbalanced classes and does not work accurately when too many irrelevant features are used. To class-balance the dataset, the Synthetic Minority Oversampling Technique (SMOTE) method is used. Meanwhile, feature selection is performed using the Ensemble Filter method, which consists of Information Gain, Gain Ratio, and Relief-F methods to handle irrelevant features. Based on the test results, the application of SMOTE and Ensemble Filter in SVM classification gives the best results with an accuracy value of 85% and an AUC of 0,850. The test can prove if SMOTE on class balancing and Ensemble Filter on feature selection can improve the classification performance of the SVM method.

Keywords: *Liver, Classification, SVM, SMOTE, Ensemble Filter*

1. PENDAHULUAN

Liver adalah salah satu organ penting dalam tubuh manusia yang berperan dalam metabolisme tubuh, salah satu fungsinya melakukan dekomposisi sel darah merah (Joloudari et al., 2019). Penyakit pada *liver* ditandai dengan berbagai jenis gangguan

seperti gangguan metabolisme, hepatitis, tumor dan sirosis (Singh, Bagga and Kaur, 2020). Berdasarkan data yang diambil dari situs *American Liver Foundation*, lebih dari 100 juta orang di Amerika Serikat memilki gangguan *liver* yang bervariasi. Masih dari *American Liver Foundation* pada tahun 2020, sebanyak 51.642 orang dewasa di Amerika Serikat meninggal akibat penyakit *liver*. Karena tingginya angka pengidap, penyakit *liver* pada pasien perlu diklasifikasikan dengan baik, agar hasil klasifikasi dapat membantu tenaga medis dalam mendiagnosa awal apakah seorang pasien memiliki penyakit *liver* atau tidak (Assegie, 2021).

Beberapa penelitian terkait klasifikasi penyakit *liver* pada pasien telah dilakukan oleh para peneliti, seperti oleh Panwar et al. (2021), menggunakan metode klasifikasi *Logistic Regression*, *Decision Tree*, SVM, *Naïve Bayes* dan *Random Forest* dimana hasilnya SVM mendapatkan nilai *accuracy* tertinggi sebesar 74,09%. Penelitian yang dilakukan oleh Assegie. (2021), menggunakan metode SVM dan K-*Nearst Neighbor* (K-NN), hasilnya SVM mendapatkan nilai *accuracy* rata-rata tertinggi sebesar 74,52% dan *area under curve* (AUC) lebih tinggi dibanding K-NN. Penelitian oleh Musyaffa and Rifai. (2018), menggunakan metode SVM dengan *Particle Swarm Optimization* (PSO), hasil tertinggi diperoleh PSO+SVM dengan *accuracy* 77,36% dan AUC 0,661.

Berdasarkan beberapa penelitian terdahulu, SVM mendapatkan nilai *accuracy* dan AUC tertinggi dalam klasifikasi penyakit *liver* pada pasien. SVM sendiri memiliki keunggulan yaitu bekerja lebih baik dibanding metode klasifikasi lain pada *dataset* bersampel kecil, tidak linier dan memiliki kelas biner (Tao, Sun and Sun, 2018). Meski begitu, SVM memiliki kelemahan ketika diterapkan pada *dataset* dengan kelas yang tidak seimbang karena sulit mendapatkan *hyperplane* pemisah optimal (Cervantes et al., 2020; Huang et al., 2021) dan tidak bekerja dengan akurat ketika terlalu banyak fitur yang tidak relevan bagi metode klasifikasi digunakan (Hamid et al., 2021; Ferdinand and Al Maki, 2022).

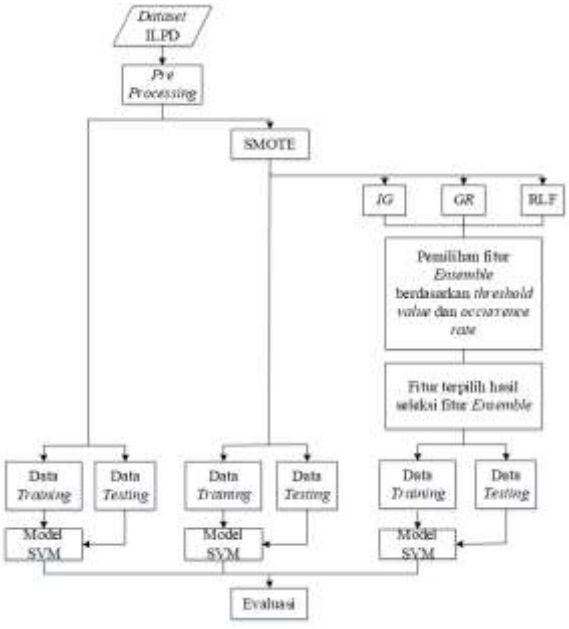
Dalam menangani masalah ketidakseimbangan kelas pada *dataset*, teknik *oversampling* lebih banyak digunakan dibandingkan *undersampling*. Alasannya, *undersampling* berpotensi menghilangkan informasi penting pada kelas mayoritas (Rahmawan and SN, 2020). Salah satu metode *oversampling* yang optimal adalah *Synthetic Minority Oversampling Technique* (SMOTE) (Ubaidillah et al., 2022). SMOTE digunakan untuk meningkatkan jumlah data kelas minoritas. Hasilnya, data kelas minoritas memiliki jumlah yang lebih seimbang dengan data kelas mayoritas (Ramadhanti, Kusuma and Annisa, 2020).

Selain masalah *dataset* dengan kelas tidak seimbang, banyaknya fitur yang tidak relevan bagi metode klasifikasi perlu ditangani. Untuk menanganinya dilakukanlah seleksi fitur (Urbanowicz et al., 2018). Metode Filter merupakan metode seleksi fitur yang direkomendasikan, karena fleksibel (bisa digunakan untuk metode klasifikasi apapun) dan sederhana (evaluasi relevansi fitur berbasis peringkat). Namun, metode Filter tunggal hanya fokus mengevaluasi fitur secara individual dan tidak mempertimbangkan pengaruh antar fitur yang dipilih, sehingga hasil seleksi kurang optimal (Hamid et al., 2021). Menurut Hamid et al. (2021), menggabungkan (*Ensemble*) beberapa metode Filter tunggal dapat menyeleksi fitur tidak relevan dan memberikan hasil seleksi lebih optimal. *Ensemble* Filter yang digunakan terdiri dari metode *Information Gain* (IG), *Gain Ratio* (GR), dan *Relief-F* (RLF), metode-metode tersebut direkomendasikan karena efisiensi komputasi dan evaluasi peringkatnya yang sederhana.

Berdasarkan paparan sebelumnya, pada penelitian ini 3 skenario percobaan dibuat. Pertama adalah klasifikasi dengan SVM, kedua adalah klasifikasi dengan SMOTE+SVM dan ketiga adalah klasifikasi dengan SMOTE+*Ensemble* Filter+SVM. SMOTE digunakan untuk menyeimbangkan kelas dari *dataset* dan *Ensemble* Filter untuk menyeleksi fitur dari *dataset* pasca SMOTE. Dengan melakukan pengujian dan evaluasi, performa keseluruhan skenario percobaan dapat diketahui.

2. METODE PENELITIAN

Dalam penelitian ini, bagan alur penelitian diperlukan untuk menggambarkan skenario penelitian yang dilakukan. Bagan alur penelitian dapat dilihat pada gambar 2.



Gambar 2. Alur penelitian

Untuk mengetahui lebih detail mengenai skenario percobaan yang terdapat pada gambar 2 dapat dilihat pada tabel 1.

Tabel 1. Skenario Percobaan

No	Nama	Skenario
1	Skenario 1	Klasifikasi SVM
2	Skenario 2	Klasifikasi SMOTE+SVM
3	Skenario 3	Klasifikasi SMOTE+*Ensemble* Filter+SVM

2.1 *Dataset*

Dataset yang digunakan adalah *Indian Liver Patient Dataset* (ILPD) dari UCI *Machine Learning Repository* yang dapat diunduh pada laman https://www.kaggle.com/datasets/uciml/indian-liver-patient-records. *Dataset* terdiri dari 10 fitur, 2 label kelas dan 583 baris data. Fitur-fiturnya adalah *Age*, *Gender*, *Total_Bilirubin, Direct_bilirubin, Alkaline_ Phosphotase, Alamine_Aminotransferase, Aspartate_Aminotransferase, Total_Protiens, Albumin* dan *Albumin_and_Globulin_Ratio.*

Adapun nilai label kelasnya adalah 1 untuk pasien dan 2 untuk bukan pasien. Rasio label kelas *dataset* adalah 416 baris data untuk kelas = 1 dan 167 baris data untuk kelas = 2.

2.2 *Pre-processing* Data

Pre-processing data dilakukan untuk memastikan data dapat diolah dengan baik. Tahapan dalam *pre-processing* diantaranya sebagai berikut. Tahap pertama adalah mengisi *missing value* dengan nilai yang paling sering muncul pada keseluruhan data yang berada pada fitur dimana *missing value* ditemukan. Tahap kedua adalah melakukan normalisasi data untuk menyamakan rentang nilai antar fitur bertipe kontinu (numerik/angka). Metode normalisasi yang digunakan adalah *z-score*. Rumus dari *z-score* dapat dilihat pada formula 1.

$$x_i^1 = \frac{x_i - \underline{A}}{\sigma_A} \quad (1)$$

Tahap ketiga adalah mentransformasi tipe data dari fitur kategorik (huruf/*string*) menjadi numerik (angka) dan mengubah label kelas yang sebelumnya berupa angka biasa (kelas = 1 dan kelas = 2) menjadi angka biner (kelas = 1 dan kelas = 0) dengan *label encoding* (Ubaidillah et al., 2022). Tujuan melakukan *label encoding* adalah untuk memudahkan proses penambangan data, karena secara umum metode-metode dalam data *mining* lebih dapat membaca data bertipe angka (Hajar, Setiawan and Bachtiar, 2022) (Santoso, Putri and Sahbandi, 2023).

2.3 SMOTE

Menerapkan SMOTE untuk menyeimbangkan jumlah data kelas minoritas dengan data kelas mayoritas pada *dataset* yang telah di *pre-processing*. SMOTE sendiri meningkatkan jumlah baris data dengan menghasilkan data sintetis acak untuk kelas minoritas dari tetangga terdekatnya menggunakan jarak *Euclidean*. Baris data baru menjadi serupa dengan data asli karena dibuat berdasarkan fitur asli *dataset* (Ishaq et al., 2021). Menurut Ubaidillah et al. (2022), rumus *Euclidean* dapat dilihat pada formula 2.

$$D(x,y) = \sqrt{(X_1 - Y_1)^2 + .. + (X_n - Y_n)^2} \quad (2)$$

Setelah menghitung jarak *instance* dengan jarak *Euclidean*, dilakukan pembuatan data replikasi dari *instance* terdekat dengan formula 3.

$$Xsyn = X_i + (X_{knn} - X_i) \times \sigma \quad (3)$$

Hasil penerapan SMOTE berupa *dataset* yang memiliki jumlah data yang seimbang untuk kedua kelasnya.

2.4 *Ensemble* Filter

Dataset dengan kelas yang sudah diseimbangkan kemudian diseleksi fiturnya. Seleksi fitur *ensemble* bertujuan untuk mendapatkan *subset* fitur yang optimal berdasarkan penggabungan hasil seleksi fitur metode tunggal yang ada didalamnya (Wang et al., 2019). Menggunakan gabungan hasil seleksi fitur dari beberapa metode tunggal didapatkanlah fitur-fitur relevan yang mengumpulkan keuntungan, meminimalkan bias dan mengompensasi kerugian dari metode-metode tunggal (Mera-Gaona et al., 2021).

Menurut Bommert et al. (2020) salah satu metode seleksi fitur *ensemble* yang direkomendasikan adalah *ensemble* Filter, terdiri dari metode Filter tunggal seperti IG, GR dan RLF. Berikut penjelasan dari masing-masing metode.

2.4.1 *Information Gain* (IG)

IG menentukan relevansi sebuah fitur dengan menghitung perolehan informasi antara fitur dengan label kelas sehingga tingkat kebergantungannya dapat diukur. Untuk mendapatkan skor peringkat, IG mengevaluasi ukuran nilai *entropy* disetiap fitur sebagai skor relevansinya. IG tertinggi setara dengan nilai *entropy* terkecil dimana suatu fitur dianggap relevan jika memperoleh nilai IG tinggi (Hamid et al., 2021).

2.4.2 *Gain Ratio* (GR)

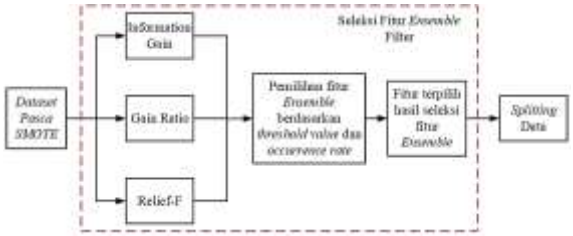
Gain Ratio (GR) adalah peningkatan dari IG, GR bekerja untuk meningkatkan bias dari IG terhadap fitur dengan perbedaan nilai yang tinggi (Dai and Xu, 2013). GR menentukan relevansi fitur dengan mengevaluasi signifikansi informasi menggunakan ukuran nilai *entropy*. Berdasarkan mekanisme percabangan, informasi yang tersebar merata menghasilkan nilai GR yang lebih tinggi, sedangkan informasi yang tidak merata menghasilkan nilai GR yang lebih kecil (Hamid et al., 2021).

2.4.3 *Relief-F* (RLF)

RLF menghitung relevansi fitur menggunakan pengujian berkelanjutan untuk mengevaluasi perbedaan bobot fitur pada kelas yang sama (*hits* terdekat) dan kelas yang berbeda (*miss* terdekat). Fitur signifikan dipilih berdasarkan kemampuannya untuk memisahkan *instance* dari kelas yang berbeda. Artinya, fitur dengan skor lebih tinggi diindikasikan dengan bobot fitur lebih tinggi di kelas yang sama (*hits* terdekat tinggi) dan fitur dengan skor lebih rendah diindikasikan dengan bobot fitur lebih tinggi di kelas berbeda (*miss* terdekat tinggi). Fitur relevan adalah fitur dengan nilai RLF yang tinggi (Urbanowicz et al., 2018; Hamid et al., 2021).

2.4.4 Tahapan Seleksi Fitur *Ensemble* Filter

Menurut Hamid et al. (2021) tahapan *ensemble* Filter dibagi menjadi 3. Tahap 1 adalah melakukan penilaian pada setiap fitur menggunakan 3 metode Filter tunggal. Tahap 2 adalah melakukan seleksi fitur secara *ensemble*, yakni melakukan pemeringkatan fitur dari hasil penilaian metode Filter tunggal. Pemeringkatan dilakukan berdasarkan *threshold value* dan *occurrence rate* yang ditentukan. Tahap 3 adalah mendapatkan *subset* fitur optimal terpilih hasil *ensemble* yang dapat digunakan untuk tahapan klasifikasi. Gambar 2 merupakan bagan alur proses *ensemble* Filter.



Gambar 2. Alur proses seleksi fitur *Ensemble* Filter

2.5 *Splitting* Data

Dataset dengan *subset* fitur optimal hasil seleksi fitur digunakan untuk membangun model klasifikasi SVM. Sebelum digunakan, *dataset* di *split* menjadi 2 bagian. *Splitting* data dilakukan secara acak menggunakan metode *Stratified sampling* dengan perbandingan data *training* 80% dan data *testing* 20%. Rasio pembagian data 80%:20% sendiri umum digunakan, karena memberikan hasil yang cukup baik dalam kebanyakan kasus terutama kasus dengan jumlah data yang tidak terlalu besar (Santoso, Putri and Sahbandi, 2023). Rasio 80%:20% juga mengikuti penelitian terdahulu yang dilakukan oleh Panwar et al. (2021) dan Assegie. (2021).

2.6 Klasifikasi SVM

Model klasifikasi dibangun menggunakan data *training*, metode klasifikasi yang digunakan adalah SVM. SVM merupakan metode klasifikasi untuk data linier dan tidak linier. SVM bekerja dengan memetakan data *training,* kemudian mencari *hyperplane* pemisah yang optimal berdasarkan *support vector* (Awalina, Bachtiar and Indriati, 2022) (Han, Kamber and Pei, 2011). Pada penelitian ini fungsi kernel *Radial Basis Function* (RBF) digunakan, kernel ini terdiri dari parameter regularisasi (C) dan parameter fungsi/*gamma* (γ). C dikenal sebagai parameter penalti biaya yang mengidentifikasi biaya *trade-off* antara mengurangi kesalahan pelatihan dan kompleksitas model, sedangkan γ menentukan pemetaan *hyperplane* tidak linier dari ruang *input* ke dalam ruang fitur berdimensi tinggi. Nilai-nilai parameter yang digunakan adalah C=8 dan γ=2 (Hamid et al., 2021). Menurut Thaseen and Kumar. (2017), rumus dari fungsi kernel RBF dapat dilihat pada formula 4.

$$k(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right) \quad (4)$$

Model yang sudah dibangun selanjutnya diuji dengan data *testing*.

2.7 Evaluasi Performa Model Klasifikasi

Model klasifikasi yang telah diuji selanjutnya dievaluasi untuk mengetahui performanya. Menurut Putri, Nugroho and Herteno. (2021), Ramadhanti, Kusuma and Annisa. (2020), dan Han, Kamber and Pei. (2011), dalam evaluasi terdapat beberapa indikator pengukuran yaitu *accuracy*, *recall, precision*, *specificity*, *area under the* ROC (*Receiver Operating Characteristic*) *curve* (AUROC atau AUC) dan *F1-score*. Berikut penjelasan dari masing-masing indikator.

2.7.1 *Accuracy*

Accuracy digunakan untuk mengetahui persentase keseluruhan kelas data yang diklasifikasi dengan benar. Rumus *accuracy* dapat dilihat pada formula 5.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (5)$$

2.7.2 *Recall*

Recall/True Positive Rate (TPrate) digunakan untuk mengetahui persentase data dengan kelas positif yang diklasifikasi sebagai positif. Rumus *recall* dapat dilihat pada formula 6.

$$Recall/TPrate = \frac{TP}{TP+FN} \quad (6)$$

2.7.3 *Precision*

Precision digunakan untuk mengetahui persentase data yang diklasifikasi sebagai kelas positif dengan benar. Rumus *precision* dapat dilihat pada formula 7.

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

2.7.4 *Specificity*

Specificity digunakan untuk mengetahui persentase data yang diklasifikasi sebagai kelas negatif dengan benar. Rumus *specificity* dapat dilihat pada formula 8.

$$Specificity = \frac{TN}{TN+FP} \quad (8)$$

2.7.5 AUC

AUC digunakan untuk membedakan kinerja model dengan memberitahu kemampuan model dalam mengklasifikasikan kelas positif dan negatif. Rumus AUC dapat dilihat pada formula 9.

$$AUC = \frac{1+TPrate-FPrate}{2} \quad (9)$$

Nilai *TPrate* didapatkan menggunakan formula 6 dan *FPrate* menggunakan formula 10.

$$FPrate = \frac{FP}{FP+TN} \quad (10)$$

2.7.6 *F1-score*

F1-*score* digunakan untuk menggabungkan *precision* dan *recall* menjadi satu indikator pengukuran. F1-*score* menghitung performa kelas minoritas secara menyeluruh untuk menghindari masalah ketika TP dan FP meningkat secara bersamaan. Rumus dari F1-*score* dapat dilihat pada formula 11.

$$F1-score = \frac{2(precisionxrecall)}{(precision+recall)} \quad (11)$$

3. HASIL DAN PEMBAHASAN

Pengujian untuk 3 skenario percobaan yang terdapat pada tabel 1 dilakukan. Hasil yang didapat dijelaskan pada sub-bab 3.2 sampai 3.5.

3.1 Analisis Hasil *Pre-Processing* Data

Dataset yang digunakan memiliki 4 baris data kosong atau *missing value* pada fitur *Albumin_and_Globulin_Ratio,* maka nilai tersebut diisi menggunakan nilai yang paling sering muncul pada kolom fiturnya atau modus (Joloudari et al., 2019). Selain *missing value, dataset* memiliki rentang nilai antar fitur kontinu yang beragam. Untuk memudahkan proses penambangan data, *dataset* dinormalisasi agar rentang nilai antar fiturnya seragam. Pada tabel 2 dapat dilihat jika rentang nilai pada fitur *Age* berbeda dengan nilai pada fitur *Albumin.* Adapun *dataset* asli tanpa *pre-processing* juga dapat dilihat pada tabel 2.

Normalisasi data dilakukan menggunakan *z-score* karena metode ini dapat mengatasi kelemahan dari metode seperti *min-max* atau penskalaan desimal (Suyanto, 2019). Berikutnya, data pada fitur *Gender* yang bernilai "*Female*" ditransformasikan menjadi angka 1 dan "*Male*" ditransformasikan menjadi angka 0. Terakhir, label kelas pada *dataset* yang terdiri dari angka 1 dan 2 ditransformasikan menjadi angka 1 untuk kelas=1 dan 0 untuk kelas=2. *Dataset* hasil *pre-processing* dapat dilihat pada tabel 3.

Tabel 2. Sampel *dataset* asli sebelum di *pre-processing*

Age	*Gender*	*Total_Bilirubin*	*Direct_Bilirubin*	*Alkaline_Phosphotase*	*Alamine_Aminotransferase*	*Aspartate_Aminotransferase*	*Total_Protiens*	*Albumin*	*Albumin_and_Globulin_Ratio*	*Dataset (kelas)*
65	*Female*	0,7	0,1	187	16	18	6,8	3,3	0,9	1
62	*Male*	10,9	5,5	699	64	100	7,5	3,2	0,74	1
62	*Male*	7,3	4,1	490	60	68	7	3,3	0,89	1
58	*Male*	1	0,4	182	14	20	6,8	3,4	1	1
72	*Male*	3,9	2	195	27	59	7,3	2,4	0,4	1
…	…	…	…	…	…	…	…	…	…	…
60	*Male*	0,5	0,1	500	20	34	5,9	1,6	0,37	2
40	*Male*	0,6	0,1	98	35	31	6	3,2	1,1	1
52	*Male*	0,8	0,2	245	48	49	6,4	3,2	1	1
31	*Male*	1,3	0,5	184	29	32	6,8	3,4	1	1
38	*Male*	1	0,3	216	21	24	7,3	4,4	1,5	2

Tabel 3. Sampel *dataset* hasil *pre-processing*

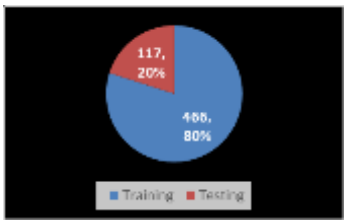
Age	*Gender*	*Total_Bilirubin*	*Direct_Bilirubin*	*Alkaline_Phosphotase*	*Alamine_Aminotransferase*	*Aspartate_Aminotransferase*	*Total_Protiens*	*Albumin*	*Albumin_and_Globulin_Ratio*	*Dataset* (kelas)
1,252098	1	-0,418877832	-0,493963976	-0,4267149 6	-0,354665405	-0,318393333	0,292119608	0,1989687	-0,147678089	1

Age	*Gender*	*Total_Bilirubin*	*Direct_Bilirubin*	*Alkaline_Phosphotase*	*Alamine_Aminotransferase*	*Aspartate_Aminotransferase*	*Total_Proteins*	*Albumin*	*Albumin_and_Globulin_Ratio*	*Dataset* (kelas)
1,066637	0	1,225171351	1,430423338	1,682628563	-0,091599335	-0,034332575	0,937566343	0,0731566	-0,649880502	1
1,066637	0	0,644918698	0,931508108	0,821587945	-0,113521507	-0,145185554	0,476532961	0,1989687	-0,17906574	1
0,819356	0	-0,370523445	-0,387053569	-0,447314017	-0,365626491	-0,311465022	0,292119608	0,3247807	0,166198418	1
1,684839	0	0,096902304	0,183135264	-0,393756467	-0,294379431	-0,176362954	0,753152 99	-0,93334	-1,717060629	1
…	…	…	…	…	…	…	…	…	…	…
0,942997	0	-0,451114091	-0,493963976	0,862786061	-0,332743233	-0,262966844	-0,537740 48	1,939837	-1,811223581	0
-0,29341	0	-0,434995962	-0,493963976	-0,793378189	-0,250535086	-0,273359 31	-0,445533804	0,0731566	-0,480074926	1
0,448435	0	-0,402759703	-0,458327173	-0,187765889	-0,179288025	-0,211004 51	-0,076707098	0,0731566	-0,166198418	1
-0,84979	0	-0,322169057	-0,351416767	-0,439074394	-0,283418345	-0,269895155	0,292119608	0,3247807	-0,166198418	1
-0,41705	0	-0,370523445	-0,422690371	-0,307240424	-0,327262 69	-0,297608 4	0,753152 99	1,582901 5	-1,735580958	0

Setelah di *pre-processing, dataset* di *split* menjadi 2 bagian yakni data *training* dan *testing* dengan rasio perbandingan 80%:20% (Ubaidillah et al., 2022). Pembagian sampel data dilakukan menggunakan *Stratified sampling. Stratified sampling* digunakan agar data dari masing-masing kelas terbagi secara merata pada partisi *training* maupun *testing* (Mahmud et al., 2020). Kemudian skenario-skenario percobaan seperti pada tabel 1 diuji.

3.2 Analisis Performa Model SVM

Skenario 1 adalah membangun model klasifikasi dengan metode SVM menggunakan *dataset* yang telah di *pre-processing*. Rasio dan jumlah data yang digunakan dapat dilihat pada gambar 3.



Gambar 3. Data *training* dan *testing*

Jumlah data dari masing-masing kelas dapat diihat pada tabel 4.

Tabel 4. Jumlah data berdasarkan kelasnya

Kelas	Data *Training*	Data *Testing*
0	133	34
1	333	83

Untuk mengukur performa dari model, digunakanlah *confusion matrix* hasil pengujian yang dapat dilihat pada tabel 5.

Tabel 5. *Confusion matrix* model SVM

		Predicted	
		Positif =1	**Negatif =0**
Actual	Positif =1	76	7
	Negatif =0	18	16

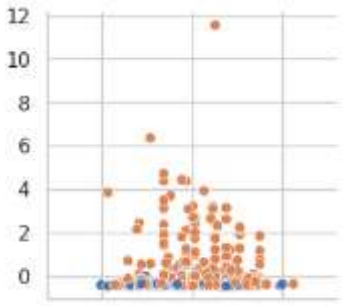
Berdasarkan *confusion matrix* pada tabel 5, didapatkan nilai evaluasi performa seperti *accuracy, precision, recall, specificity,* AUC, dan *F1-Score* yang dapat dilihat pada gambar 4.

Gambar 4. Grafik nilai performa dari model SVM untuk klasifikasi penyakit *liver*

Berdasarkan gambar 4, nilai performa yang didapatkan oleh model klasifikasi SVM untuk klasifikasi penyakit *liver* diantaranya *accuracy* 79%, *precision* 81%, *recall* 92%, *specificity* 47%, f1-*score* 86% dan AUC 0,693.

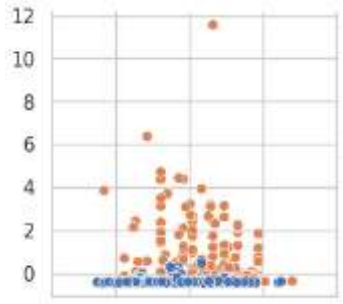
3.3 Analisis Performa Model SMOTE+SVM

Skenario 2 adalah membangun model klasifikasi SVM menggunakan *dataset* yang sudah diseimbangkan kelasnya dengan SMOTE. *Dataset* asli memiliki jumlah kelas 1 (positif) sebanyak 416 baris data dan kelas 0 (negatif) sebanyak 167 baris data. Pada gambar 5 dapat dilihat persebaran data dari kedua kelas pada fitur *Total_Billirubin.* Titik berwarna jingga adalah kelas 1 dan biru adalah kelas 0.

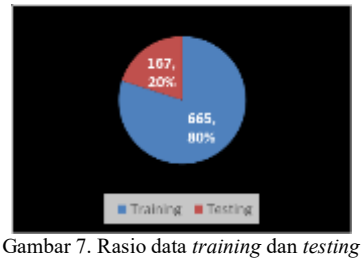


Gambar 5. Persebaran data pada fitur *Total_Billiruin*

Pada gambar 5 dapat dilihat jika sebaran data dengan kelas 1 lebih banyak dari data dengan kelas 0, maka untuk menyeimbangkan persebarannya dilakukanlah *oversampling* dengan SMOTE. Hasil penerapan SMOTE dapat dilihat pada gambar 6 yang merupakan sebaran data dari fitur *Total_Billirubin* pasca SMOTE diterapkan.



Gambar 6. Persebaran data pada fitur *Total_Billirubin* pasca SMOTE



Gambar 7. Rasio data *training* dan *testing*

Oversampling oleh SMOTE tidak hanya berlaku di fitur *Total_Billirubin,* namun ke semua fitur yang ada dalam *dataset.* Maka didapatkan *dataset* dengan jumlah data kelas 1 (positif) sebanyak 416 baris dan kelas 0 (negatif) sebanyak 416 baris. Alhasil jumlah data di keseluruhan kelas *dataset* menjadi seimbang.

Setelah penyeimbangan kelas dilakukan, *dataset* hasil SMOTE digunakan untuk klasifikasi. Rasio dan jumlah data yang digunakan dapat dilihat pada gambar 7.
Jumlah data dari masing-masing kelas dapat diihat pada tabel 6.

Tabel 6. Jumlah data berdasarkan kelasnya

Kelas	Data *Training*	Data *Testing*
0	332	84
1	333	83

Pengukuran performa dari model klasifikasi menggunakan *confusion matrix* hasil pengujian yang dapat dilihat pada tabel 7.

Tabel 7. *Confusion matrix* model SMOTE+SVM

		Predicted	
		Positif (1)	**Negatif (0)**
Actual	Positif (1)	65	18
	Negatif (0)	11	73

Berdasarkan *confusion matrix* pada tabel 7, didapatkan nilai evaluasi performa seperti *accuracy, precision, recall, specificity,* AUC dan *F1-Score* yang dapat dilihat pada gambar 8.

Gambar 8. Grafik nilai performa dari model SMOTE+SVM untuk klasifikasi penyakit *liver*

Berdasarkan gambar 8, nilai performa yang didapatkan oleh model klasifikasi SMOTE+SVM untuk klasifikasi penyakit *liver* diantaranya *accuracy* 83%, *precision* 86%, *recall* 80%, *specificity* 87%, f1-*score* 83% dan AUC 0,832.

3.4 Analisis Performa Model SMOTE+Ensemble Filter+SVM

Skenario 3 adalah membangun model klasifikasi SVM menggunakan *dataset* yang telah diseimbangkan kelasnya menggunakan SMOTE seperti pada sub-bab 3.3. *Dataset* hasil SMOTE kemudian diseleksi fiturnya untuk mendapatkan *subset* fitur yang relevan menggunakan *Ensemble* Filter. Sebelum seleksi fitur diterapkan, *dataset* memiliki jumlah fitur sebanyak 10, yang dapat dilihat rinciannya pada tabel 8.

Tabel 8. Fitur-fitur pada *dataset*

No.	Nama Fitur
1.	*Age*
2.	*Gender*
3.	*Total_Bilirubin*

No.	Nama Fitur
4.	*Direct_Bilirubin*
5.	*Alkaline_Phospotase*
6.	*Alamine_Aminotransferase*
7.	*Aspartate_Aminotransferase*
8.	*Total_Protein*
9.	*Albumin*
10.	*Albumin_and_Globulin_Ratio*

Mengacu pada gambar 2, proses penyeleksian fitur dengan *Ensemble* Filter dikerjakan dalam 3 tahapan. Tahap 1 adalah penilaian fitur pada masing-masing metode Filter tunggal, yaitu IG, GR, dan RLF. Untuk hasil penilaian dari ketiga metode dapat dilihat pada tabel 9.

Tabel 9. Hasil penilaian fitur pada metode Filter tunggal

	IG	No. Fitur	GR	No. Fitur	RLF	No. Fitur
Nilai & *Rank*	0,156	4	0,156	4	0,269	3
	0,155	3	0,142	3	0,219	4
	0,124	6	0,105	7	0,202	9
	0,122	7	0,085	5	0,108	7
	0,1	5	0,082	6	0,099	10
	0,046	10	0,062	10	0,094	2
	0,022	9	0,055	1	0,065	5
	0,022	1	0,023	9	0,025	6
	0,001	2	0,001	2	0	8
	0	8	0	8	-0,049	1

Setelah didapatkan nilai dan peringkat fitur dari masing-masing metode, tahap 2 yakni seleksi fitur *Ensemble* dilakukan. Proses ini menggunakan *threshold value* = 0,05 dan *occurrence rate* = sejumlah metode yang digunakan, yaitu 3. Penentuan nilai-nilai tersebut mengikuti penelitian terdahulu oleh (Hamid et al., 2021). Fitur dengan nilai dibawah *threshold value* pada seluruh metode tunggal dieliminasi. Jika fitur memiliki nilai diatas *threshold value* pada salah satu atau lebih metode tunggal, fitur tetap digunakan untuk pemilihan berdasarkan *occurrence rate.* Proses seleksi fitur *ensemble* dapat dilihat pada tabel 10.

Tabel 10. Tahap seleksi fitur *ensemble* dan perhitungan *occurrence of rate*

Rank	No. Fitur IG	GR	RLF	*Top Dominan*	*Occurrence of Rate* 1	2	3
1	4	4	3	4	✓	✓	✓
2	3	3	4	3	✓	✓	✓
3	6	7	9	7	✓	✓	✓
4	7	5	7	6	✓	✓	-
5	5	6	10	9	✓	-	-
6	10	10	2	5	✓	✓	✓
7	9	1	5	10	✓	✓	-
8	1	9	6	2	✓	-	-
9	2	2	8	1	✓	-	-
10	8	8	1	8	-	-	-

Pada tabel 10 dapat diketahui jika fitur bertuliskan merah, maka fitur tersebut dieliminasi. Fitur yang ditulis hitam adalah fitur yang memiliki nilai diatas *threshold value.* Sedangkan fitur yang bertuliskan biru adalah fitur yang nilainya berada dibawah *threshold value* namun tidak dieliminasi karena pada salah satu atau lebih metode tunggal nilainya berada diatas *threshold value.* Selanjutnya fitur diurutkan berdasarkan urutan *rank* nya pada seluruh metode tunggal, sehingga didapatkan *Top Dominan* fitur. Lalu *occurrence of rate* dihitung, jika suatu fitur nilainya berada diatas *threshold value* pada 3 metode tunggal, maka dihitung sebagai 3. Untuk fitur yang nilainya berada diatas *threshold value* pada 2 metode tunggal maka dihitung sebagai 2. Untuk fitur yang nilainya diatas *threshold value* pada satu metode tunggal saja dihitung sebagai 1.

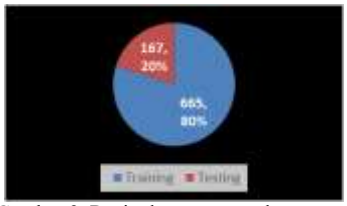
Berdasarkan *occurrence of rate,* didapatkanlah *subset* fitur optimal baru dari *dataset*. Tahap 3, *Subset* fitur optimal yang didapat adalah sejumlah nilai *occurrence of rate* yaitu 3 yang dapat dilihat pada tabel 11.

Tabel 11. *Subset* fitur optimal hasil *ensemble* Filter

Optimal 1	Optimal 2	Optimal 3
4	4	4
3	3	3
7	7	7
6	6	5
9	5	
5	10	
10		
2		
1		

Subset fitur optimal kemudian digunakan dalam proses klasifikasi. Sesuai hasil seleksi fitur, *dataset* dibuat menjadi 3 yaitu *dataset* optimal 1, *dataset* optimal 2, dan *dataset* optimal 3. Masing-masing *dataset* baru memiliki jumlah kelas yang sama (kelas yang sudah diseimbangkan) namun terdiri dari fitur berbeda sesuai hasil seleksi fitur.

Rasio dan jumlah data yang digunakan dapat dilihat pada gambar 9.



Gambar 9. Rasio data *training* dan *testing*

Jumlah data dari masing-masing kelas dapat diihat pada tabel 12.

Tabel 12. Jumlah data berdasarkan kelasnya

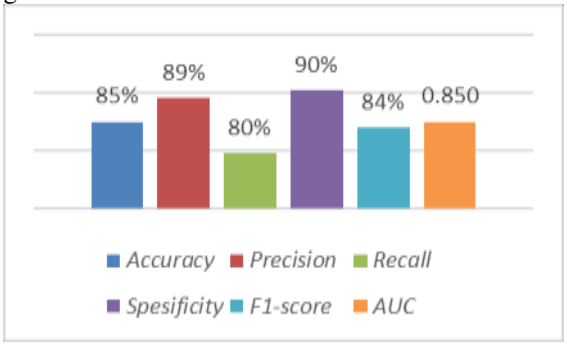
Kelas	Data *Training*	Data *Testing*
0	332	84
1	333	83

Performa dari model klasifikasi SVM yang dibangun berdasarkan masing-masing *dataset* optimal diukur menggunakan nilai *accuracy*. Nilai *accuracy* yang didapatkan oleh masing-masing *dataset* optimal dapat dilihat pada tabel 13.

Tabel 13. Nilai *accuracy* model SMOTE+*Ensemble* Filter+SVM dengan 3 *dataset* optimal

Dataset	Jumlah Fitur	*Accuracy*
Optimal 1	9	85%
Optimal 2	6	73%
Optimal 3	4	71%

Berdasarkan hasil pengujian pada tabel 13, *accuracy* terbaik didapatkan oleh *dataset* dengan *subset* fitur optimal 1 yang terdiri dari 9 fitur. Maka *dataset* dengan *subset* fitur optimal 1 dievaluasi lebih lanjut menggunakan indikator pengkuran yang ada pada sub bab 2.7. Hasil evaluasi dapat dilihat pada gambar 10.



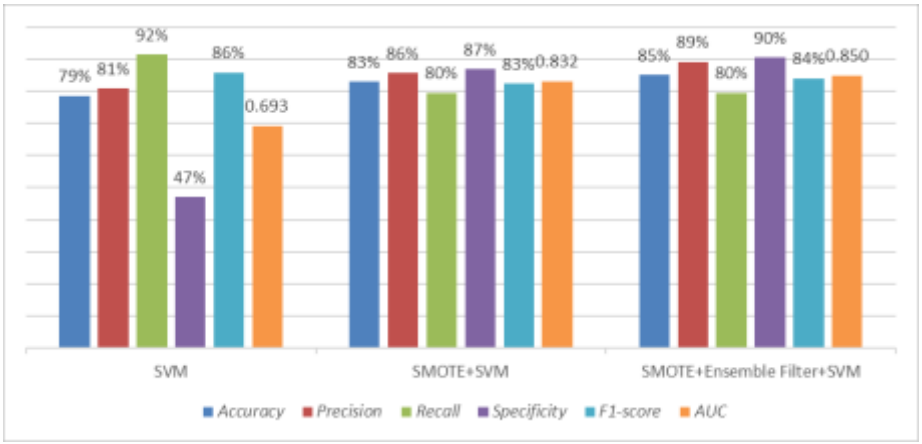
Gambar 10. Grafik nilai performa dari model SMOTE+*Ensemble* Filter+SVM untuk klasifikasi penyakit *liver*

Berdasarkan gambar 10, nilai performa yang didapatkan oleh model klasifikasi SMOTE+*Ensemble* Filter+SVM untuk klasifikasi penyakit *liver* diantaranya *accuracy* 85%, *precision* 89%, *recall* 80%, *specificity* 90%, f1-*score* 84% dan AUC 0,850.

3.5 Perbandingan Performa

Setelah semua skenario percobaan diuji dan dievaluasi performanya, keseluruhan nilai hasil evaluasi dibandingkan untuk mengetahui apakah terjadi peningkatan atau penurunan performa pada model untuk klasifikasi penyakit *liver*. Pada gambar 11 dapat dilihat keseluruhan nilai hasil evaluasi performa dari ketiga skenario percobaan.

Berdasarkan grafik perbandingan keseluruhan skenario percobaan pada gambar 11, dapat diketahui jika SMOTE+*Ensemble* Filter+SVM mendapatkan nilai *accuracy* tertinggi sebesar 85%. Nilai tersebut lebih tinggi 2% dari SMOTE+SVM dan 6% dari SVM. Nilai *precision* tertinggi juga didapatkan oleh SMOTE+*Ensemble* Filter+SVM dengan nilai 89%, meningkat sebesar 3% dari SMOTE+SVM dan 7% dari SVM Pada nilai *recall,* SVM mendapatkan nilai tertinggi yaitu 92%, lebih tinggi 12% dari SMOTE+SVM dan SMOTE+*Ensemble* Filter+SVM. Pada *specificity* SMOTE+*Ensemble* Filter+SVM mendapatkan nilai tertinggi yaitu 90% yang lebih tinggi 43% dari SVM dan 3% dari SMOTE+SVM. F1-*score* tertinggi didapatkan oleh SVM sebesar 86% lebih tinggi 3% dari SMOTE+SVM dan 2% dari SMOTE+*Ensemble* Filter+SVM. Terakhir ada nilai AUC, pada AUC, SMOTE+*Ensemble* Filter+SVM mampu mendapat nilai tertinggi 0,850, lebih tinggi dari SVM dengan nilai 0,693 maupun SMOTE+SVM 0,826.



Gambar 11. Grafik perbandingan nilai performa dari klasifikasi penyakit *liver* menggunakan SVM, SMOTE+SVM, dan SMOTE+*Ensemble* Filter+SVM

4. KESIMPULAN

Berdasarkan hasil pengujian dan evaluasi performa dari model klasifikasi yang dibangun, metode SMOTE dan *Ensemble* Filter mampu menangani masalah yang dimiliki oleh metode SVM. SMOTE dapat menangani masalah ketidakseimbangan kelas dengan membuat data sintetis bagi kelas minoritas sehingga jumlah kelas yang awalnya minoritas menjadi seimbang dengan kelas lainnya. Sedangkan metode *Ensemble* Filter dapat menyeleksi fitur *dataset* sehingga mendapatkan fitur-fitur yang relevan bagi proses klasifikasi. Adapun untuk model klasifikasi penyakit *liver* terbaik pada penelitian ini adalah SMOTE+*Ensemble* Filter+SVM yang berhasil mendapatkan nilai tertinggi pada *accuracy* 85%, *precision* 89%, *specificity* 90%, dan AUC 0,850. Nilai tersebut lebih tinggi dibandingkan nilai pada model SVM dan SMOTE+SVM.

DAFTAR PUSTAKA

ASSEGIE, T.A., 2021. Support Vector Machine and K-Nearest Neighbor Based Liver Disease Classification Model. Indonesian Journal of Electronics, Electromedical, and Medical Informatics (IJEEEMI), [online] 3(1), pp.9–14. Available at: <http://ijeeemi.poltekkesdepkes-sby.ac.id/index.php/ijeeemi>.

AWALINA, A., BACHTIAR, F.A. & INDRIATI, 2022. Klasifikasi Ulasan Palsu Menggunakan Borderline Over-Sampling (Bos) Dan Support Vector Machine (Svm) (Studi Kasus: Ulasan Tempat Makan) Spam Review Classification Using Borderline Over-Sampling And Support Vector Machine Algorithm. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 9(2), pp.419–426. https://doi.org/10.25126/jtiik.202295692.

BOMMERT, A., SUN, X., BISCHL, B., RAHNENFÜHRER, J. & LANG, M., 2020. Benchmark for Filter Methods for Feature Selection in High-Dimensional Classification Data. Computational Statistics & Data Analysis, [online] 143. https://doi.org/10.1016/j.csda.2019.106839.

CERVANTES, J., GARCIA-LAMONT, F., RODRÍGUEZ-MAZAHUA, L. & LOPEZ, A., 2020. A Comprehensive Survey on Support Vector Machine Classification: Applications, Challenges and Trends. Neurocomputing. [online] https://doi.org/10.1016/j.neucom.2019.10.118.

DAI, J. & XU, Q., 2013. Attribute Selection Based on Information Gain Ratio in Fuzzy Rough Set Theory with Application to Tumor Classification. Applied Soft Computing, 13(1), pp.211–221. https://doi.org/10.1016/j.asoc.2012.07.029.

FERDINAND, Y. & AL MAKI, W.F., 2022. Broccoli Leaf Diseases Classification Using Support Vector Machine with Particle Swarm Optimization based on Feature Selection. International Journal of Advances in Intelligent Informatics, 8(3), pp.337–348. https://doi.org/10.26555/ijain.v8i3.951.

HAJAR, N., SETIAWAN, N.Y. & BACHTIAR, F.A., 2022. Pengelompokan Mahasiswa untuk Pengajuan Bantuan Uang Kuliah Tunggal menggunakan Metode K-Means Clustering (Studi Kasus BEM FILKOM UB). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, [online] 6(5), pp.2353–2361. Available at: <http://j-ptiik.ub.ac.id>.

HAMID, T.M.T.A., SALLEHUDDIN, R., YUNOS, Z.M. & ALI, A., 2021. Ensemble Based Filter Feature Selection with Harmonize Particle Swarm Optimization and Support Vector Machine for Optimal Cancer Classification. Machine Learning with Applications, 5. https://doi.org/10.1016/j.mlwa.2021.100054.

HAN, J., KAMBER, M. & PEI, J., 2011. Data Mining Concepts and Techniques. Third Edition ed. Waltham: Morgan Kaufmann Publisher.

HUANG, B., ZHU, Y., WANG, Z. & FANG, Z., 2021. Imbalanced Data Classification Algorithm Based on Clustering and SVM. Journal of Circuits, Systems and Computers, 30(2). https://doi.org/10.1142/S0218126621500365.

ISHAQ, A., SADIQ, S., UMER, M., ULLAH, S., MIRJALILI, S., RUPAPARA, V. & NAPPI, M., 2021. Improving the Prediction of Heart Failure Patients Survival Using SMOTE and Effective Data Mining Techniques. IEEE Access, 9, pp.39707–39716. https://doi.org/10.1109/ACCESS.2021.3064084.

JOLOUDARI, J.H., SAADATFAR, H., DEHZANGI, A. & SHAMSHIRBAND, S., 2019. Computer-aided Decision-making for Predicting Liver Disease Using PSO-based Optimized SVM with Feature Selection. Informatics in Medicine Unlocked, 17. https://doi.org/10.1016/j.imu.2019.100255.

MAHMUD, M.S., HUANG, J.Z., SALLOUM, S., EMARA, T.Z. & SADATDIYNOV, K., 2020. A survey of data partitioning and sampling methods to support big data analysis. Big Data Mining and Analytics, 3(2), pp.85–101. https://doi.org/10.26599/BDMA.2019.9020015.

MD, A.Q., KULKARNI, S., JOSHUA, C.J., VAICHOLE, T., MOHAN, S. & IWENDI, C., 2023. Enhanced Preprocessing Approach Using Ensemble Machine Learning Algorithms for Detecting Liver Disease.

Biomedicines, [online] 11. https://doi.org/10.3390/biomedicines11020581.

MERA-GAONA, M., LÓPEZ, D.M., VARGAS-CANAS, R. & NEUMANN, U., 2021. Framework for the Ensemble of Feature Selection Methods. Applied Sciences, 11. https://doi.org/10.3390/app11178122.

MUSYAFFA, N. & RIFAI, B., 2018. Model Support Vector Machine Berbasis Particle Swarm Optimization untuk Prediksi Penyakit Liver. JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER, 3(2).

PANWAR, V., CHOUDHARY, N., MITTAL, S. & SAHU, G., 2021. Review of Liver Disease Prediction using Machine Learning Algorithm. Journal of Emerging Technologies and Innovative Research (JETIR), [online] 8(2). Available at: <www.jetir.org>.

PUTRI, N.L., NUGROHO, R.A. & HERTENO, R., 2021. Intrusion Detection System Berbasis Seleksi Fitur dengan Kombinasi Filter Information Gain Ratio dan Correlation. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 8(3), pp.457–464. https://doi.org/10.25126/jtiik.202183154.

RAHMAWAN, H. & SN, A., 2020. Penentuan Rekomendasi Pelatihan Pengembangan Diri bagi Pegawai Negeri Sipil Menggunakan Algoritma C4.5 dengan Principal Component Analysis dan Diskritisasi. Jurnal TEKNO KOMPAK, 14(1), pp.5–10.

RAMADHANTI, N.S., KUSUMA, W.A. & ANNISA, 2020. Optimasi Data Tidak Seimbang pada Interaksi Drug Target dengan Sampling dan Ensemble Support Vector Machine. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 7(6), pp.1221–1230. https://doi.org/10.25126/jtiik.202072857.

SANTOSO, H., PUTRI, R.A. & SAHBANDI, 2023. Deteksi Komentar Cyberbullying pada Media Sosial Instagram Menggunakan Algoritma Random Forest. Jurnal Manajemen Informatika (JAMIKA), 13(1). https://doi.org/10.34010/jamika.v13i1.9303.

SINGH, J., BAGGA, S. & KAUR, R., 2020. Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques. In: Procedia Computer Science. Elsevier B.V. pp.1970–1980. https://doi.org/10.1016/j.procs.2020.03.226.

SUN, Y., QUE, H., CAI, Q., ZHAO, J., LI, J., KONG, Z. & WANG, S., 2022. Borderline SMOTE Algorithm and Feature Selection-Based Network Anomalies Detection Strategy. Energies, 15. https://doi.org/10.3390/en15134751.

SUYANTO, 2019. Data Mining Untuk Klasifikasi Dan Klasterisasi Data Edisi Revisi. Bandung: Informatika.

TAO, P., SUN, Z. & SUN, Z., 2018. An Improved Intrusion Detection Algorithm Based on GA and SVM. IEEE Access, 6, pp.13624–13631. https://doi.org/10.1109/ACCESS.2018.2810198.

THASEEN, I.S. & KUMAR, C.A., 2017. Intrusion detection model using fusion of chi-square feature selection and multi class SVM. Journal of King Saud University - Computer and Information Sciences, 29(4), pp.462–472. https://doi.org/10.1016/j.jksuci.2015.12.004.

UBAIDILLAH, R., MULIADI, NUGRAHADI, D.T., FAISAL, M.R. & HERTENO, R., 2022. Implementasi XGBoost pada Keseimbangan Liver Patient Dataset dengan SMOTE dan Hyperparameter Tuning Bayesian Search. Jurnal Media Informatika Budidarma, 6(3), pp.1723–1729. https://doi.org/10.30865/mib.v6i3.4146.

URBANOWICZ, R.J., MEEKER, M., LA CAVA, W., OLSON, R.S. & MOORE, J.H., 2018. Relief-based Feature Selection: Introduction and Review. Journal of Biomedical Informatics, 85, pp.189–203. https://doi.org/10.1016/j.jbi.2018.07.014.

WANG, J., XU, J., ZHAO, C., PENG, Y. & WANG, H., 2019. An Ensemble Feature Selection Method for High-dimensional Data Based on Sort Aggregation. Systems Science & Control Engineering, [online] 7(2), pp.32–39. https://doi.org/10.1080/21642583.2019.1620658.

Halaman ini sengaja dikosongkan.