DISERTASI IF186601

SISTEM DETEKSI INTRUSI MENGGUNAKAN METODE KLASIFIKASI BERBASIS CENTROID DAN SUB-CENTROID TETANGGA TERDEKAT

BAMBANG SETIAWAN
NRP. 05111560010005

Dosen Pembimbing
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.
Tohari Ahmad, S.Kom., MIT., Ph.D.

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2020

Halaman ini sengaja dikosongkan

PENGESAHAN

Disertasi disusun untuk memenuhi salah satu syarat memperoleh gelar
Doktor Ilmu Komputer (Dr.)
di
Institut Teknologi Sepuluh Nopember

oleh:
Bambang Setiawan
NRP. 05111 5600 10005

Tanggal ujian : 13 Januari 2020
Periode wisuda : Maret 2020

Disetujui oleh:

1. Prof. Ir. Supeno Djanali, M.Sc., Ph.D. .....
   NIP. 19480619 197301 1 001 (Pembimbing 1)

2. Tohari Ahmad, S.Kom., MIT., Ph.D. .....
   NIP. 19750525 200312 1 002 (Pembimbing 2)

3. Prof. Dr. Ir. Richardus Eko Indrajit, M.Sc., MBA. .....
   NIDN. 0324016902 (Penguji Eksternal)

4. Waskitho Wibisono, S.Kom., M. Eng, Ph.D. .....
   NIP. 19741022 200003 1 001 (Penguji 1)

5. Dr. Royyana Muslim Ijtihadie, S.Kom.,M.Kom. .....
   NIP. 19770824 200604 1 001 (Penguji 2)

Kepala Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas

Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
NIP. 19751220 200112 2 002

.

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Disertasi saya dengan judul:

SISTEM DETEKSI INTRUSI MENGGUNAKAN METODE KLASIFIKASI BERBASIS CENTROID DAN SUB-CENTROID TETANGGA TERDEKAT

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan **bukan** merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 30 Januari 2020

Bambang Setiawan
NRP. 05111 5600 10005

Halaman ini sengaja dikosongkan

SISTEM DETEKSI INTRUSI MENGGUNAKAN METODE KLASIFIKASI BERBASIS CENTROID DAN SUB-CENTROID TETANGGA TERDEKAT

Nama mahasiswa : Bambang Setiawan
NRP : 05111 5600 10005
Pembimbing I : Prof. Ir. Supeno Djanali, M.Sc., Ph.D.
Pembimbing II : Tohari Ahmad, S.Kom., MIT.,Ph.D.

ABSTRAK

Pendeteksian intrusi dalam lalu lintas jaringan komputer menjadi tantangan bagi para peneliti selama bertahun-tahun. Kemajuan di bidang pembelajaran mesin memberikan kesempatan kepada peneliti untuk mendeteksi intrusi jaringan tanpa menggunakan basis data *signature*. Keseimbangan antara kecepatan dan ketepatan merupakan fokus utama dari sistem deteksi intrusi, dimana aspek ketepatan ditinjau dari ukuran *accuracy* dan *completeness* dalam mendeteksi serangan. Jumlah data pelatihan pada setiap jenis serangan yang tidak berimbang dapat menyebabkan sistem deteksi intrusi memiliki *accuracy* yang tinggi tetapi sulit untuk mengenali semua jenis serangan, sehingga aspek *completeness* tidak terpenuhi.

Sudah banyak model deteksi intrusi yang dikembangkan menggunakan teknik pembelajaran mesin, baik menggunakan algoritma tunggal maupun *hybrid*, tetapi umumnya masih menghasilkan nilai *false negative rate* dan *false positive rate* yang masih tinggi serta belum dapat mendeteksi semua jenis serangan. Hal tersebut juga terjadi pada model deteksi intrusi L-SCANN yang berbasis *centroid-based classification*. Metode *centroid-based classification* merupakan salah satu bentuk pendekatan *hybrid machine learning* yang spesifik untuk meningkatkan kecepatan proses klasifikasi, dengan menggunakan dua komponen yaitu algoritma pengklaster dan algoritma pengklasifikasi. Untuk meningkatkan ketepatan deteksi L-SCANN, dalam penelitian ini dibangun pendekatan baru model deteksi intrusi dengan menggabungkan sejumlah metode.

Pada tahapan praproses, metode normalisasi Log serta metode seleksi fitur MRIGFS dan RWIGFS diajukan untuk menangani masalah *imbalanced-class*. Optimasi nilai parameter k dari L-SCANN untuk meningkatkan ketepatan prediksi. Selanjutnya penggabungan L-SCANN dengan SVM-OP dan SVM-OW dengan teknik *ensemble-voting* dilakukan untuk memvalidasi prediksi *false negative* dan *false positive*. Dimana SVM-OP adalah SVM dengan optimasi kernel RBF, dan SVM-OW merupakan *cost learning* SVM dengan optimasi bobot kelas.

Pengujian kinerja model dilakukan menggunakan dataset NSL-KDD dan dataset Kyoto2006++. Hasil uji coba dengan dataset NSL-KDD menunjukkan

bahwa penerapan *ensemble-voting* menggunakan SVM-OP dan SVM-OW dapat meningkatkan kinerja deteksi dari L-SCANN. A*ccuracy*, *sensitivity*, dan *specificity* ditingkatkan sampai di atas 99.0%, FPR dan FNR diturunkan sampai di bawah 0.3%, serta *sensitivity* pada *minority class* R2L dan U2R meningkat menjadi 94% dan 65%. Sedangkan hasil uji coba dengan dataset Kyoto2006++ menghasilkan *accuracy*, *sensitivity*, dan *specificity* di atas 99.0%, dengan FPR dan FNR di bawah 0.25%.

Kata kunci: *centroid-based classification*, dataset Kyoto2006++, dataset NSL-KDD, *ensemble-voting*, normalisasi Log, seleksi fitur, sistem deteksi intrusi, *support vector machine*

INTRUSION DETECTION SYSTEM USING CLASSIFICATION METHOD BASED ON CENTROID AND NEAREST NEIGHBOR SUB-CENTROID

By	: Bambang Setiawan
Student Identity Number	: 05111 5600 10005
Supervisor	: Prof. Ir. Supeno Djanali, M.Sc., Ph.D.
Co-Supervisor	: Tohari Ahmad, S.Kom., MIT.,Ph.D.

ABSTRACT

Detection of intrusions in computer network traffic has been a challenge for researchers for years. Machine learning provide an opportunity for researchers to detect network intrusion without using a signature database. The balance between speed and accuracy is the main focus of intrusion detection systems, where the aspect of accuracy in addition to prioritizing high sensitivity also considers completeness in detecting attacks. The amount of training data on each type of attack that is not balanced can cause intrusion detection systems to have high accuracy, but it is difficult to recognize all types of attacks, so the completeness aspects are not met.

Many intrusion detection models are developed using machine learning techniques, both using a single algorithm and hybrid. These models generally produce false-negative rates and false-positive rates that are high and have not able to detect all types of attacks. This conditions also occur in the L-SCANN, intrusion detection model based on centroid-based classification. Centroid-based classification method is a form of hybrid machine learning approach that is specific to increase the speed of the classification process, using two components, the clustering algorithm and the classification algorithm. To improve the accuracy of L-SCANN detection, a new approach to intrusion detection models was built in this study by combining a number of methods.

At the preprocessing stage, and two feature selection methods (MRIGFS and RWIGFS) and the Log normalization method are proposed to deal with imbalanced-class problems. Optimize the k parameter value from L-SCANN to improve the accuracy. Furthermore, the merging of L-SCANN with SVM-OP and SVM-OW with ensemble-voting technique is done to validate false-negative and false-positive predictions. Where SVM-OP is SVM with RBF kernel optimization, and SVM-OW is SVM cost learning with class weight optimization.

Model performance testing was performed using the NSL-KDD dataset and the Kyoto2006 ++ dataset. The results of experiment with the NSL-KDD dataset indicate that the application of ensemble-voting using SVM-OP and SVM-OW can improve the detection performance of L-SCANN. Accuracy, sensitivity,

and specificity are all increased to above 99.0%, FPR and FNR are reduced to below 0.3%, and sensitivity for minority classes R2L and U2R are increased to 94% and 65%. While the trial results with the Kyoto2006 ++ dataset produce accuracy, sensitivity, and specificity above 99%, with FPR and FNR below 0.25%.

Keywords: centroid-based classification, ensemble-voting, feature selection, intrusion detection system, Kyoto2006 ++ dataset, log normalization, NSL-KDD dataset, support vector machine

KATA PENGANTAR

Segala puji syukur bagi Allah SWT atas segala karunia, rahmat, taufik serta hidayah yang tidak henti-hentinya diberikan sehingga penulis mampu menyelesaikan disertasi dengan judul "Sistem Deteksi Intrusi Menggunakan Metode Klasifikasi berbasis Centroid dan Sub-Centroid Tetangga Terdekat".

Penulis tak lupa mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada semua pihak yang telah berperan serta dalam upaya penyelesaian disertasi ini, khususnya:

1. Prof. Ir. Supeno Djanali, Ph.D. selaku promotor dan Tohari Ahmad, S.Kom., MIT, Ph.D. selaku co-promotor yang senantiasa memberikan bimbingan dan arahan sehingga disertasi ini dapat diselesaikan dengan baik oleh penulis.
2. Prof. Dr. Ir. Richardus Eko Indrajit, M.Sc., MBA. selaku penguji eksternal, Waskitho Wibisono, S.Kom., M.Eng., Ph.D. serta Dr. Royyana Muslim Ijtihadie, S.Kom., M.Kom. selaku penguji internal yang telah memberikan begitu banyak masukan guna meningkatkan kualitas disertasi ini.
3. Ibu yang tak henti-hentinya memanjatkan doa kepada Allah SWT, serta memberikan support dan motivasi.
4. Istri tercinta, Prasasti Ekayani, SH. dan ananda Ekasetia Nur Sakinah, S.Si. yang selalu memberikan dukungan dan doanya.
5. Institut Teknologi Sepuluh Nopember Surabaya, khususnya departemen Sistem Informasi, yang telah memberikan kesempatan dan dukungan kepada penulis untuk melanjutkan jenjang pendidikan doktor.
6. Kemenristek DIKTI yang telah mendukung penulis dalam bentuk Beasiswa Program Pascasarjana Dalam Negeri (BPPDN).
7. Keluarga besar mahasiswa Program Doktor Ilmu Komputer ITS yang telah menjadi teman diskusi bagi penulis.
8. Semua pihak yang tidak dapat disebutkan satu persatu yang telah banyak memberikan bantuan hingga terselesaikannya disertasi ini.

Penulis menyadari betapapun kerasnya upaya penulis dalam memberikan kontribusi keilmuan dalam penelitian disertasi ini, namun tetap saja masih banyak kekurangan di sana-sini baik secara teoretis maupun metodologis, sehingga penulis selalu terbuka dalam menerima saran dan kritik untuk kelanjutan penelitian disertasi ini di dalam jenjang karir penelitian penulis ke depannya. Akhirnya harapan dan doa semoga hasil penelitian disertasi ini mampu memberikan manfaat secara keilmuan untuk dunia akademik.

Surabaya, 30 Januari 2020

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
PENDAHULUAN

1.1. Kondisi Keamanan Internet Saat Ini

Saat ini, penggunaan peralatan komputasi dan peralatan digital telah menyebar luas di semua sektor. Berbagai sarana komunikasi, seperti sistem komputer, smartphone, dan konektivitas nirkabel serta inovasi lainnya, telah menyebabkan peningkatan ketergantungan masyarakat pada peralatan tersebut. Masyarakat membutuhkannya untuk dapat mengakses layanan dan aplikasi di mana saja kapan saja, untuk menyelesaikan pekerjaan mereka dengan cepat dan efisien. Dunia maya memainkan peran yang sangat penting dalam masyarakat dan ekonomi kontemporer sejak internet telah mengubah cara orang dan organisasi berkomunikasi dan melakukan bisnis secara elektronik. Pengguna, perusahaan dan pemerintah telah menjadi tergantung pada layanan internet serta perangkat dan aplikasi-aplikasi pendukung untuk melakukan kegiatan sehari-hari mereka (Portillo, 2014).

Semua perangkat, baik komputer pribadi, *server*, *tablet*, atau *smartphone*, memungkinkan untuk diretas dengan teknik penyerangan yang juga semakin canggih. Sehingga pengamanan perangkat harus dimulai dengan menghalangi aktivitas jahat yang berupaya melakukan eksploitasi dan penetrasi melalui jaringan, untuk menembus sumber daya dan sistem operasi. Seorang penyerang dapat dengan sembunyi-sembunyi berusaha untuk menemukan kerentanan dari korban yang melakukan aktivitas di internet, seperti mengunduh file, mengunggah file, mengirim email atau mengeksekusi transaksi keuangan. Salah satu tindakan yang dilakukan adalah mengirimkan skrip berbahaya untuk mengganggu perimeter keamanan perangkat jaringan yang menjadi target.

Ancaman penyerang digambarkan sebagai serangkaian peristiwa berbahaya yang mencoba mengeksploitasi prinsip-prinsip *confidentiality* (kerahasiaan), *integrity* (integritas) dan *availability* (ketersediaan), yang dikenal dalam sistem komputer sebagai segitiga CIA (Von Solms & Van Niekerk, 2013). Karena setiap serangan memiliki ciri-ciri dan pola serangan yang berbeda, maka hal

ini menjadi tantangan besar dalam proses pendeteksiannya. Setiap serangan pada jaringan dan sistem komputer dapat menyebabkan bencana yang mengganggu kebijakan keamanan komputer dari segitiga CIA (Pontarelli, Bianchi, & Teofili, 2013). Jenis serangan *denial of service* (DoS) bertujuan menghabiskan sumber daya peralatan jaringan computer. Jenis serangan ini dapat menyebabkan sumber daya komputer tidak dapat bekerja sehingga prinsip *availability* tidak terpenuhi. Sedangkan serangan jenis *malware* dan *malicious codes* membajak aliran eksekusi program, dan hal ini akan melanggar prinsip *integrity*.

Beberapa teknik dasar yang digunakan untuk melindungi keamanan komputer adalah otentikasi pengguna, enkripsi data, dan *firewall*. Kemudian dikembangkan *intrusion detection systems* (IDS), satu sistem yang berfungsi untuk memantau dan mendeteksi adanya gangguan keamanan pada sistem komputer dan jaringan komputer. Sistem ini dikembangkan menggunakan teknik analitis khusus untuk mendeteksi serangan, mengidentifikasi sumber serangan, dan memberi peringatan kepada administrator (Yuehui Chen, Abraham, & Yang, 2007).

IDS harus mampu membedakan antara aktivitas normal di jaringan dengan aktivitas serangan dan menyediakan informasi-informasi terkait serangan untuk dianalisis lebih lanjut. Berdasarkan cara melakukan deteksi, IDS dibagi menjadi dua yaitu IDS yang berbasis *signature detection (misuse-detection)* dan IDS yang berbasis *anomaly detection* (Sommer & Paxson, 2010). Pada *signature detection*, paket atau log audit dipindai untuk mencari urutan perintah atau peristiwa yang sebelumnya sudah ditentukan sebagai indikasi serangan. Sedangkan pada *anomaly detection*, IDS menggunakan pola perilaku untuk mendeteksi aktivitas berbahaya dengan cara menganalisis aktivitas masa lalu untuk mengenali apakah perilaku yang diamati merupakan aktivitas normal atau aktivitas berbahaya.

Pada awal pengembangannya, IDS banyak menggunakan *signature detection.* Namun karena metode ini menghasilkan tingkat alarm palsu (*false alarm rate*) yang tinggi, maka dikembangkan IDS berbasis *anomaly detection* yang menggunakan pendekatan inovatif baru seperti analisis statistik, teknik data mining, dan pembelajaran mesin (*machine learning*) (Sarasamma, Zhu, & Huff, 2005).

Pada McAfee Labs Threat Report bulan Desember 2017 (Minihane et al., 2017) disebutkan bahwa jumlah ancaman terhadap jaringan meningkat secara

dramatis. Hal ini menyebabkan kerugian finansial dan kerusakan reputasi, pencurian informasi sensitif dan kekayaan intelektual, termasuk gangguan di bidang kesehatan. Menurut laporan *Cost of Cyber-Crime Study* dari Accenture yang dikeluarkan pada tahun 2017 (Accenture and Ponemon Institute, 2017), selama tahun 2016-2017 biaya percepatan kejahatan dunia maya meningkat lebih dari 23 persen dibanding tahun sebelumnya. Biaya yang dikeluarkan organisasi-organisasi untuk mengelola insiden atau pengeluaran untuk pemulihan dari gangguan per tahun rata-rata US $ 11,7 juta. Karena biaya akibat kejahatan dunia maya, yang secara substansial terus meningkat, ada motivasi yang kuat bagi kami untuk melakukan penelitian dan pengembangan model sistem deteksi penyusupan (*intrusion detection system* - IDS).

Sebagian besar penelitian IDS dengan deteksi *anomaly* menggunakan berbagai variasi teknik *data mining* dan pembelajaran mesin. Dan banyak pula penelitian yang menggabungkan atau mengintegrasikan teknik yang berbeda untuk meningkatkan kinerja deteksi. Praproses data (*data pre-processing*) merupakan tahapan dalam *data mining* dan pembelajaran mesin yang berhubungan dengan upaya peningkatan kualitas data sebelum dilakukan proses pembelajaran. Kegiatan pada tahapan ini dapat dikelompokkan menjadi dua, yaitu preparasi data (*data preparation*) dan reduksi data (*data reduction*). Preparasi data meliputi integrasi data, pembersihan data, normalisasi data, dan transformasi data. Sedangkan kegiatan pada reduksi data meliputi seleksi fitur, seleksi instan, dan diskritisasi.

Menurut (García, 2015; Ogasawara et al., 2010), praproses data merupakan tahapan penting dalam teknik *data mining* dan pembelajaran mesin, tetapi sering diabaikan. Untuk mengetahui apakah kondisi yang sama juga terjadi pada bidang IDS, kami melakukan studi literatur pada penelitian-penelitian IDS berbasis pembelajaran mesin *hybrid* pada tahun 2007-2016.

Proses yang kami amati pada studi literatur meliputi: 1) konversi nilai fitur simbolik ke numerik, 2) normalisasi, 3) seleksi fitur, 4) penanganan record data yang redundan, dan 5) penanganan kondisi jumlah sampel data yang tidak seimbang pada setiap kelas (*imbalanced class*). Hal ini kami lakukan berdasar pada kondisi dataset IDS KDD-Cup 1999 yang banyak digunakan pada penelitian IDS. Berikut adalah kondisi dataset IDS KDD-Cup 1999 menurut (Tavallaee, Bagheri, Lu, &

Ghorbani, 2009): 1) memiliki banyak fitur yang terdiri dari gabungan fitur-fitur numerik dan fitur-fitur simbolik, 2) fitur-fitur numerik mempunyai rentang nilai yang beragam, 3) merupakan *imbalanced dataset*, dan 4) memiliki banyak record yang redundan.

Kami juga melakukan studi pada model-model IDS yang menggunakan fitur representatif (*representative feature*). Dimana fitur representatif dibentuk dalam upaya untuk mendapatkan sekumpulan fitur baru yang mempunyai kekuatan pembeda (*discriminative power*) lebih baik dibanding fitur-fitur asli pada dataset. Hal tersebut menarik untuk dilakukan mengingat bahwa sudah sangat banyak penelitian pada bidang IDS yang menggunakan dataset KDD-Cup 1999 atau DARPA untuk uji coba tetapi belum ada jawaban pasti fitur-fitur mana yang lebih representatif (Lin, Ke, & Tsai, 2015). Kondisi yang sama dimungkinkan akan ditemui ketika menggunakan dataset IDS yang lain.

Keseimbangan aspek kecepatan dan ketepatan merupakan poin utama dari kinerja IDS, dimana aspek ketepatan ditinjau dari ukuran *accuracy* dan *completeness* dalam mendeteksi serangan (Debar, Dacier, & Wespi, 1999; Nazer, 2011). Banyak sistem deteksi penyusupan telah dikembangkan berdasarkan teknik pembelajaran mesin menggunakan metode tunggal dan *hybrid*, tetapi tidak banyak yang membahas tentang aspek *completeness.* Jumlah data training pada setiap jenis serangan yang tidak berimbang menyebabkan sistem deteksi memiliki akurasi yang tinggi tetapi sistem deteksi sulit untuk mengenali semua jenis serangan, sehingga aspek *completeness* tidak terpenuhi.

1.2. Penelitian IDS Berbasis Pembelajaran Mesin Hybrid pada Tahun 2007-2018

Kami melakukan studi literatur pada 78 makalah penelitian IDS berbasis pembelajaran mesin *hybrid* dalam rentang waktu tahun 2007-2018 (Setiawan, Djanali, & Ahmad, 2017). Rangkuman hasil studi kami sajikan pada Tabel 1.1. Dalam tabel tersebut ditunjukkan beberapa informasi terkait IDS dan tahapan preproses data yang dilakukan, antara lain: algoritma yang digunakan pada model *hybrid*, dataset IDS yang digunakan, pembahasan *completeness*, penyeleksian fitur

(P1), penskalaan atau normalisasi fitur numerik (P2), konversi nilai fitur simbolik ke numerik (P3), penanganan kondisi *imbalanced class* (P4), dan penanganan record data yang redundan (P5).

Tabel 1.1 Penelitian IDS berbasis *Hybrid machine learning* Tahun 2007-2018

no	Tahun	Penulis	Dataset	Metode yang digunakan	Comple-teness	Praproses Data				
						P1	P2	P3	P4	P5
1	2018	Mahendiran dan Appusani	NSL-KDD	oneR-FS + CRF	Ya	Ya	Ya	Ya	-	-
2	2018	Kumar dkk	NSL-KDD	MLDR + multiclass SVM	Ya	Ya	Ya	Ya	-	-
3	2017	Bostani dan Sheikhan	NSL-KDD	Modified OPF	Ya	Ya	Ya	Ya	-	-
4	2017	Thaseen dan Khumar	NSL-KDD	Chi-FS + multiclass SVM	Ya	Ya	Ya	Ya	-	-
5	2017	Pajouh dkk	NSL-KDD	Naïve Bayes + CF-KNN	Ya	Ya	Ya	Ya	Ya	-
6	2017	Yaseen dkk	NSL-KDD	Hybrid SVM + ELM	Ya	Ya	Ya	Ya	-	-
7	2016	Muttaqien dkk	NSL-KDD, Kyoto 2006+	Divisive K-means + K-NN	Ya	Ya	Ya	Ya	-	Ya
8	2016	Ahmad dkk	NSL-KDD, Kyoto 2006+	Bisecting K-means+ K-NN	Ya	Ya	Ya	Ya	-	-
9	2016	Al-Jarrah dkk	ISOT botnet	K-means + DT	Tidak	Ya	-	-	-	Ya
10	2016	Aburomman dkk	KDD-Cup 1999	PCA + SVM	Tidak	Ya	Ya	Ya	-	-
11	2016	Corrales dkk	KDD-Cup 1999	C4.5 - (SVM/MLP/SVM)	-	-	-	-	Ya	Ya
12	2016	Milliken	NSL-KDD, ISCX 2012	GA, Base Level Classifier	Tidak	-	-	-	-	-
13	2015	Kharisma dkk	NSL-KDD, Kyoto 2006+	Recursive K-means + K-NN	Ya	Ya	Ya	Ya	-	Ya
14	2015	Lin dkk	KDD-Cup 1999	K-means + K-NN (CANN)	Ya	Ya	-	-	-	-
15	2015	Yang dan Hui	-	Ant Colony Cluster,K-means	Tidak	-	-	-	-	-
16	2015	Liang Hu dkk	DARPA 2000	K-means, Fuzzy C Mean	Tidak	-	-	-	-	-
17	2015	Eesa dkk	KDD-Cup 1999	DT	Tidak	Ya	-	-	-	-
18	2015	Koucham dkk	DARPA 1999	Hierarchical Clustering + NB	Tidak	-	-	-	-	-
19	2015	Senthilnayaki dkk	KDD-Cup 1999	SVM	Tidak	Ya	-	-	-	-
20	2015	Aissa dan Guerroumi	KDD-Cup 1999	Genetic Optimation	Tidak	Ya	Ya	Ya	-	-
21	2015	Sani dan Ghasemi	Kyoto 2006+	SVM Clustering	Tidak	Ya	Ya	-	-	-
22	2014	Abdurrazaq dkk	KDD-Cup 1999	Ant Colony Cluster	Tidak	-	-	-	-	-
23	2014	Eslamnezhad dan Varjan	NSL-KDD	Min Max K-means	Tidak	-	-	-	-	-
24	2014	Yassin dkk	DARPA 1999	Naïve baya, Random Forest	Tidak	-	-	-	-	-
25	2014	Masarat dkk	KDD-Cup 1999	Fuzzy Ensemble Classifier	Tidak	Ya	Ya	-	-	-
26	2014	E de la Hoz dkk	KDD-Cup 1999, DARPA 1999	GHSOM	Tidak	Ya	Ya	-	-	-
27	2014	Feng dkk	KDD-Cup 1999	SVM+ant colony network	Tidak	Tidak	Ya	-	-	-
28	2014	Kim dkk	KDD-Cup 1999	DT + SVM	Tidak	Tidak	Ya	-	-	-
29	2013	Elbasiony dkk	KDD-Cup 1999	weighted K-means + RForest	Tidak	-	Ya	Ya	Ya	Ya
30	2013	Chun Guo dkk	KDD-Cup 1999	CBC algorithm + KNN	Tidak	-	-	-	-	Ya
31	2013	Aljarah dan Ludwig	KDD-Cup 1999	MR-CPSO	Tidak	-	Ya	Ya	-	-
32	2013	L. Shen dan L. Feng	DARPA 1999	Rough meta-learning	Tidak	Ya	Ya	Ya	-	-
33	2013	Senthilnayaki dkk	KDD-Cup 1999	Modified J48 + Decision Tree	Tidak	Ya	-	-	-	-
34	2013	Thaseen dan Kumar	NSL-KDD	Supervised Tree Classifier	Tidak	Ya	Ya	-	-	-
35	2013	Baig dkk	KDD-Cup 1999	GMDH	Tidak	Ya	Ya	-	-	-
36	2013	Shin dkk	DARPA 2000	Markov chain + probabilistic	Tidak	Ya	Ya	-	-	-
37	2012	Muniyandi dkk	DARPA 1999	K-Means + C4.5 DT	Tidak	Ya	-	-	-	-
38	2012	Chitrakar dan Huang	KDD-Cup 1999, Kyoto2006+	K-Medoid + Naïve Baya	Tidak	Ya	Ya	Ya	-	-
39	2012	Guorui dkk	KDD-Cup 1999	Fuzzy C-Means	Tidak	-	Ya	Ya	-	-
40	2012	Sedjelmaci	KDD-Cup 1999	Graph based cluster	Tidak	-	-	-	-	-
41	2012	Sharma	KDD-Cup 1999	K-means + Naïve Baya	Tidak	Ya	Ya	-	-	-
42	2012	H. Guo dkk	KDD-Cup 1999	Genetic Clustering	Tidak	-	Ya	Ya	-	-
43	2012	Lin dkk	KDD-Cup 1999	DT, SVM	Tidak	Ya	Ya	-	-	-
44	2011	Ming-Yang Su	online/network	MBLG + K-NN	Tidak	Ya	Ya	-	-	-
45	2011	Shi-Jinn Horng dkk	KDD-Cup 1999	hierarchical clustering +SVM	Tidak	Ya	Ya	-	-	-
46	2011	Z. Li dkk	KDD-Cup 1999	K-Means + PSO	Tidak	Ya	Ya	-	-	-
47	2011	Z. Muda dkk	KDD-Cup 1999	K-means + Naïve Baya	Tidak	Ya	Ya	-	-	-
48	2011	Ruzhi Xu	KDD-Cup 1999	PSO-RBF Classifier	Tidak	-	-	Ya	-	-
49	2011	Ghadiri dan N. Ghadiri	KDD-Cup 1999	Fuzzy Clustering + RBF NN	Tidak	-	-	Ya	-	Ya

no	Tahun	Penulis	Dataset	Hybrid Machine Learning	Complete-ness	Preprocessing				
						P1	P2	P3	P4	P5
50	2011	Ishida dkk	KDD-Cup 1999, Kyoto2006++	OptiGrid and a cluster labelling	Tidak	Ya	Ya	-	-	-
51	2011	Wei Song Li dkk	KDD-Cup 1999	Fuzzy C-Means + ACO	Tidak	Ya	Ya	Ya	-	-
52	2011	Muda dkk	KDD-Cup 1999	K-Means + OneR	Tidak	Ya	Ya	Ya	-	-
53	2011	Zhong dkk	KDD-Cup 1999, Kyoto2006++	Grid-Based Clustering	Tidak	Ya	Ya	Ya	-	-
54	2010	K. Q. Yan dkk	KDD-Cup 1999	CSWN+BPN	Tidak	Ya	Ya	Ya	-	-
55	2010	X. Li	KDD-Cup 1999	Multi-classifier NN	Tidak	Ya	Ya	-	-	-
56	2010	Mi dan L. Hai	KDD-Cup 1999	Giacinto's multiple classifiers	Tidak	Ya	Ya	-	-	-
57	2010	Teng dkk	KDD-Cup 1999	fuzzy c-means clustering	Tidak	Ya	Ya	-	-	-
58	2010	Bahrbegi	KDD-Cup 1999	GA-based clustering	Tidak	Ya	Ya	-	-	-
53	2010	Wang	KDD-Cup 1999	Fuzzy C-Mean + QPSO	Tidak	Ya	Ya	Ya	-	-
60	2010	Kenaza dan Zaidi	NSL-KDD	K-means + Naïve Baya	Tidak	Ya	Ya	Ya	-	-
61	2010	Tsai dan Lin	NSL-KDD	K-means + K-NN	Tidak	Ya	Ya	Ya	-	-
62	2010	Sangkatsanee dkk	KDD-Cup 1999	DT, ANN, Ripper rule	Tidak	Ya	Ya	Ya	-	-
63	2010	Wang dkk	KDD-Cup 1999	Fuzzy clustering + ANN	Tidak	Tidak	Ya	-	-	-
64	2009	Tajbakhsh dkk	KDD-Cup 1999	FL+AR	Tidak	Tidak	Ya	-	-	-
65	2009	Tong dkk	DARPA 1999	RBF, Elman neural network	Tidak	Tidak	Ya	-	-	-
66	2008	Giacinto dkk	KDD-Cup 1999	k-means, SVM ensembles	Tidak	Tidak	Ya	Ya	-	-
67	2008	Hu dkk	KDD-Cup 1999	AdaBoost DT	Tidak	Tidak	Ya	Ya	-	-
68	2008	Xiang dkk	KDD-Cup 1999	Baya clustering + DT	Tidak	Ya	Ya	Ya	-	-
69	2007	Abadeh dkk	DARPA 1998	GA+FL	Tidak	Tidak	Ya	-	-	-
70	2007	Chen dkk	DARPA 1998	GA + ANN	Tidak	Ya	Ya	-	-	-
71	2007	Hansen dkk	KDD-Cup 1999	GA	Tidak	Tidak	Ya	-	-	-
72	2007	Khan dkk	DARPA 1998	SOM + SVM	Tidak	Tidak	Ya	Ya	-	-
73	2007	Li dan Guo	KDD-Cup 1999	TCM k-NN	Tidak	Tidak	Ya	Ya	-	-
74	2007	Liu dkk	DARPA 1998	SOM + ANN	Tidak	Ya	Ya	Ya	-	-
75	2007	Ozyer dkk	KDD-Cup 1999	GA + FL	Tidak	Tidak	Ya	-	-	-
76	2007	Peddabachigari dkk	KDD-Cup 1999	DT + SVM	Tidak	Tidak	Ya	Ya	-	-
77	2007	Shon dan Moon	DARPA 1999	GA + SVM	Tidak	Ya	Ya	-	-	-
78	2007	Kayacik dkk	KDD-Cup 1999	SOM	Tidak	Tidak	Ya	Ya	-	-

****Catatan:*
Completeness: membahas aspek Completeness, P1: Seleksi Fitur, P2: Normalisasi, P3: Konversi nilai fitur simbolik ke numerik, P4: Menangani kondisi imbalanced class, P5:Menangani record data yang redundan

Sebagian besar dari literatur tersebut menggunakan dataset IDS DARPA 1999 atau dataset pengembangannya seperti KDD-Cup 1999 dan NSL-KDD. Dataset tersebut merupakan *imbalanced dataset* dan memiliki banyak record yang redundan (Tavallaee et al., 2009). Kedua kondisi ini sangat mempengaruhi kinerja klasifikasi dan meningkatkan waktu proses. Sedangkan kondisi *imbalanced class* akan membuat pengklasifikasi mengalami kesulitan dalam mendeteksi *minority class.*

Tabel 1.1 juga menunjukkan bahwa kondisi tahapan praproses data juga kurang mendapat perhatian pada penelitian IDS. Tidak semua literatur menginformasikan secara jelas tahapan praproses data yang dilakukan dalam penelitian. Dari 78 makalah IDS tersebut di atas, hanya 34 yang melakukan konversi nilai fitur simbolik ke numerik, sebagian besar menggunakan metode *arbitrary*. Sebanyak 60 penelitian sudah melakukan proses normalisasi dan

sebagian besar menggunakan metode min-max. Hanya 3 penelitian yang melakukan penanganan kondisi *imbalanced class* dan ketiganya menggunakan teknik *resampling.* Proses penyeleksian fitur sudah dilakukan oleh sebagian besar penelitian yaitu sebanyak 47 penelitian. Namun hanya 7 penelitian yang melakukan penghapusan record data yang redundan, dan hanya 10 penelitian yang membahas aspek *completeness*.

Berikut adalah 10 penelitian yang membahas aspek *completeness*: (Ahmad & Muchammad, 2016; Al-Yaseen, Othman, & Nazri, 2017; Bostani & Sheikhan, 2017; Guo et al., 2014; Kumar, Raju, & Vardhan, 2018; Lin et al., 2015; Mahendiran & Appusamy, 2018; Muchammad & Ahmad, 2015; Muttaqien & Ahmad, 2017; Pajouh, Dastghaibyfard, & Hashemi, 2017; Sumaiya Thaseen & Aswani Kumar, 2017). Dari 10 penelitian tersebut, empat penelitian yang menggunakan menggunakan metode *Centroid-based Classification* (CBC), yaitu (Ahmad & Muchammad, 2016; Lin et al., 2015; Muchammad & Ahmad, 2015; Muttaqien & Ahmad, 2017).

Metode CBC merupakan salah satu bentuk pendekatan *hybrid machine learning* yang spesifik untuk meningkatkan kecepatan proses klasifikasi, dengan menggunakan dua komponen yaitu algoritma pengklaster (*clustering*) dan algoritma pengklasifikasi (Lin et al., 2015). Proses perhitungan dalam tahap pelatihan dan klasifikasi hanya didasarkan pada pusat-pusat klaster (*centroid)* dan tidak melibatkan seluruh data pelatihan sehingga waktu proses lebih pendek. Ditinjau dari sudut kualitas dataset, metode ini melakukan upaya peningkatan kualitas dataset dengan membentuk fitur representatif untuk diproses oleh pengklasifikasi. Dimana fitur representatif dibentuk dari jarak satu titik data ke *centroid* dan jarak dari satu titik data ke titik data terdekatnya dalam satu klaster yang sama akan memiliki kekuatan diskriminatif yang lebih baik untuk mendeteksi kesamaan kelas dari titik data yang diuji (C. F. Tsai & Lin, 2010). Pada bagian selanjutnya, kami akan membahas perkembangan penelitian IDS yang menggunakan metode CBC yang menjadi fokus dalam penelitian ini.

1.3. Perkembangan Penelitian IDS Berbasis Centroid-based Classification

Penelitian IDS berbasis CBC diawali oleh Chih-Fong Tsai dan Jung-Hsiang Tsai yang membuat IDS berbasis CBC dengan menggunakan pembentukan RF berbasis *centroid* dan *nearest neighbor* yang dikenal dengan CANN (C.-F. Tsai, Tsai, & Chou, 2012). Chun Guo dkk selanjutnya membuat IDS berbasis CBC dengan menggunakan pembentukan fitur representatif berbasis centroid dan melakukan proses klasifikasi dengan menggunakan SVM, metode ini dikenal dengan metode DSSVM (Guo et al., 2014).

Pengembangan fitur representatif berbasis *centroid* dan *sub-centroid* diawali dengan penelitian (Muchammad & Ahmad, 2015) yang kemudian disempurnakan pada (Ahmad & Muchammad, 2016). Metode ini dikenal dengan nama *L-SCANN - logarithmic subcentroid and nearest neighbor*. Pengembangan IDS berbasis CBC yang terbaru mencoba memperbaiki kinerja dari metode L-SCANN dengan menggantikan *sub-centroid* dengan *sub-medoid* (Muttaqien & Ahmad, 2017). Metode ini kami sebut dengan *CASMN -Centroid and SubMedoid Neighbor*. Penggunaan jarak ke *centroid* sebagai dasar pembentukan fitur representatif dilakukan dalam upaya untuk mengurangi waktu komputasi yang dibutuhkan pada saat pelatihan dan pendeteksian. Dengan alasan, bahwa *centroid* merupakan rata-rata jarak antara dua titik data yang ada dalam klaster, maka *centroid* dapat dianggap sebagai representasi dari semua titik data yang ada dalam klaster. Alasan yang sama juga berlaku pada penggunaan *sub-centroid* atau *sub-medoid*.

Penelitian DSSVM, CANN, L-SCANN, dan CASMN lebih banyak membahas proses pengklasteran dan pembangkitan fitur representatif, namun proses klasifikasi dan praproses data belum dibahas lebih jauh. Penelitian-penelitian tersebut hanya menginformasikan tentang penggunaan metode *dimentional reduction* atau *feature selection*, namun kurang menginformasikan tentang metode untuk normalisasi dan penanganan *imbalanced class*. Penelitian DSSVM menginformasikan penggunaan metode min-max untuk normalisasi, sedangkan penelitian lainnya tidak memberikan informasi. Dan keempat penelitian tersebut tidak ada yang memberikan informasi tentang penanganan kondisi

imbalanced class dari dataset yang digunakan. Pada pengujian model, keempat penelitian tersebut menggunakan dataset NSL-KDD 20% dengan metode 10-*fold cross validation* untuk pelatihan dan pengujian.

Pada Tabel 1.2 kami menyajikan perbandingan kinerja deteksi dari keempat IDS berbasis CBC tersebut. Pada tabel tersebut terlihat bahwa IDS berbasis CBC mempunyai kinerja deteksi yang tinggi tetapi masih kesulitan dalam mendeteksi serangan jenis U2R dan R2L yang merupakan *minority class* pada dataset KDD-Cup 1999 dan NSL-KDD.

Tabel 1.2 Perbandingan Kinerja dengan model IDS berbasis CBC

Metode IDS	**∑ fitur**	**Accuracy (%)**					
		Overall	**Normal**	**DoS**	**R2L**	**Probe**	**U2R**
DSSVM (Guo et al., 2014)	52	92.50	98.40	97.21	***6.31***	87.49	***3.07***
CANN (Lin et al., 2015)	19	99.46	97.04	99.68	57.05	87.61	***3.85***
L-SCANN (Ahmad & Muchammad, 2016)	19	95.54	97.50	96.90	88.35	79.43	***18.18***
CASMN (Muttaqien & Ahmad, 2017)	19	96.87	97.43	97.70	48.57	91.22	***9.09***

Kami mempunyai hipotesa bahwa penyebab rendahnya kinerja deteksi model IDS berbasis CBC terhadap serangan yang tergolong *minority class*, khususnya U2R, adalah karena model IDS belum melakukan penanganan dengan baik pada tahapan praproses data dan belum dilakukannya optimasi parameter algoritma pengklasifikasi yang digunakan. Tahapan praproses data yang perlu ditangani lebih jauh adalah proses normalisasi maupun proses penanganan kondisi *imbalanced class* dari dataset IDS yang digunakan. Dengan melihat poin utama dari kinerja IDS adalah keseimbangan dari kecepatan dan ketepatan, pada penelitian ini kami akan menggunakan pendekatan penanganan *imbalanced class* pada tahapan seleksi fitur dan menggunakan metode normalisasi yang mampu mendukung peningkatan kecepatan dan ketepatan.

1.4. Perumusan Permasalahan

Dari latar belakang dan pokok permasalahan pada penelitian IDS berbasis CBC diatas, maka permasalahan pada penelitian ini kami rumuskan sebagai berikut:

a. Bagaimana menggunakan seleksi fitur untuk menangani kondisi *imbalanced class* pada dataset IDS.
b. Bagaimana menggunakan metode normalisasi untuk meningkatkan kecepatan deteksi, meningkatkan ketepatan deteksi, serta menghindari perubahan nilai *mutual information* dari fitur yang diproses.
c. Bagaimana melakukan optimasi parameter pada proses klasifikasi untuk mendapatkan kondisi *completeness*.
d. Bagaimana membangun model IDS berbasis CBC yang mampu melakukan *validasi* terhadap serangan yang diprediksi sebagai lalu lintas jaringan normal atau kondisi *False Negative.*

1.5. Batasan Masalah

Dalam penelitian ini, IDS yang dirancang merupakan IDS forensik yang dioperasikan secara off-line. IDS memproses data lalulintas jaringan yang sudah dikumpulkan pada periode waktu tertentu. Uji coba model IDS dilakukan menggunakan dataset NSL-KDD dan dataset Kyoto2006++.

Kondisi *completeness* pada penelitian ini direpresentasikan dengan kondisi *overall accuracy* yang tinggi dengan jumlah *false negative* yang rendah, dan *sensitivity* yang tinggi pada *minority class.*

1.6. Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah membangun model IDS melalui kombinasi proses normalisasi, seleksi fitur untuk *imbalanced class*, dan penggabungan tiga pengklasifikasi yaitu Centroid-based Classification (CBC), SVM dengan optimasi parameter kernel RBF (SVM-OP), dan SVM dengan optimasi bobot kelas (SVM-OW) menggunakan pendekatan *ensemble-voting*.

Kami membandingkan dan memilih satu dari tiga metode normalisasi (Min-max, Zscore, dan Log) yang unggul dalam hal kecepatan dan ketepatan deteksi, serta mampu menghindari perubahan nilai *mutual information* dari fitur

yang diproses. Pada proses seleksi fitur, kami mengajukan dua metode seleksi fitur untuk *imbalanced class* yaitu *Modified Rank Information Gain Feature Selection* (MRIGFS) dan *Rank Weighted Information Gain Feature Selection* (RWIGFS) yang bertujuan untuk mendapatkan fitur-fitur yang lebih mendukung deteksi terhadap *minority class*.

Manfaat dari penelitian ini adalah:

a. Tersedianya metode seleksi fitur untuk *imbalanced-class* yang mampu menghasilkan fitur-fitur yang mendukung deteksi *minority-class.*
b. Tersedianya metode normalisasi yang lebih mendukung kecepatan dan ketepatan hasil klasifikasi, serta mampu menghindari perubahan *mutual information* dari fitur yang diproses akibat pembulatan hasil normalisasi dengan digit kecil.
c. Tersedianya model IDS berbasis ensemble-voting yang mampu melakukan *validasi* terhadap serangan yang diprediksi sebagai lalu lintas jaringan normal atau kondisi *False Negative.*
d. Penggunaan ensemble-voting menggunakan pasangan SVM-OP dan SVM-OW untuk validasi kondisi *False Negative* dapat digeneralisasi untuk digunakan pada pengklasifikasi dan permasalahan yang lain.

1.7. Kontribusi Penelitian

Kontribusi penelitian ini adalah:

a. Metode seleksi fitur untuk *imbalanced class* yang mendukung peningkatan kemampuan fitur dalam mendeteksi *minority class*.
b. Pendekatan baru IDS dengan SVM-*Optimation Class Weight* untuk meningkatkan deteksi pada *minority class* dan mempertahankan kinerja deteksi kelas lainnya tetap tinggi.
c. Pendekatan baru menggunakan pasangan SVM-*Optimation Class Weight* dan SVM-*Optimation Kernel Parameter* untuk meminimalkan *false negative rate* dan *false positive rate* dari suatu pengklasifikasi melalui *ensemble voting*.

Halaman ini sengaja dikosongkan

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

2.1. Kajian Pustaka Penelitian IDS Berbasis Pembelajaran Mesin Hybrid yang Membahas Aspek Completeness

Pada sub-bagian ini kami membahas beberapa penelitian IDS pada tahun 2014-2018 yang membahas aspek *completeness*, yang menyajikan akurasi deteksi keseluruhan dan akurasi deteksi pada masing-masing kelas serangan. Kinerja dari penelitian-penelitian tersebut akan kami jadikan sebagai pembanding dari kinerja model IDS yang diusulkan pada penelitian ini. Ringkasan kinerja dari penelitian-penelitian tersebut kami sajikan pada Tabel 2.1.

Penelitian yang pertama adalah penelitian (Guo et al., 2014) yang mengusulkan IDS berbasis CBC dengan menggunakan pembentukan RF berbasis *centroid* dan melakukan proses klasifikasi dengan menggunakan SVM, metode ini dikenal dengan metode *DSSVM – Distance Sum SVM*. Uji coba model dilakukan pada dataset NSL-KDD. Hasil uji coba menghasilkan akurasi keseluruhan 92.50%. Sedangkan *accuracy* untuk masing-masing kelas secara berturutan adalah sebagai berikut; kelas normal= 98.40%, kelas DoS= 97.21%, kelas R2L= 6.31%, kelas Probe= 87.49%, dan kelas U2R= 3.07%.

(Lin et al., 2015) mengusulkan pendekatan serupa menggunakan pengklasifikasi berbasis *centroid*, yaitu *cluster center and nearest neighbor* (CANN). Pendekatan ini menghasilkan fitur representatif satu dimensi dari penjumlahan dua jarak. Jarak pertama adalah jarak antara titik data ke semua *centroid*, sedangkan jarak kedua adalah jarak antara titik data ke tetangga terdekat dalam klaster yang sama. Klasifikasi k-NN digunakan untuk memproses fitur representatif satu dimensi. Pendekatan ini juga dapat mendeteksi semua kelas serangan. Keakuratan klasifikasi pendekatan ini untuk kelas keseluruhan lebih dari 99% tetapi *accuracy* kelas U2R masih di bawah 5%. Sedangkan *accuracy* kelas normal= 97.04%, kelas DoS= 99.68%, kelas Probe=87.49%, dan kelas R2L= 57.05%.

(Muchammad & Ahmad, 2015) dan (Ahmad & Muchammad, 2016) mengajukan metode untuk meningkatkan kinerja CANN yang diberi nama *logarithmic sub-centroid and nearest neighbor* (L-SCANN). Metode ini menggantikan jarak ke *nearest neighbor* dengan jarak ke *sub-centroid*. CASMN-*centroid and sub-medoid neighbor* (Muttaqien & Ahmad, 2017) melakukan hal yang sama dengan L-SCANN, namun *sub-centroid* diganti dengan *sub-medoid*. Kedua metode tersebut dapat meningkatkan kinerja deteksi pada kelas R2L dan U2R, namun *accuracy* pada kelas U2R masih di bawah 19%.

(Bostani & Sheikhan, 2017) menggunakan algoritma *modified optimum path forest* (OPF) untuk mendeteksi intrusi. Mereka meningkatkan kualitas dataset pelatihan dengan mempartisi data pelatihan ke dalam himpunan pelatihan yang homogen menggunakan algoritma pengklasteran k-means. Pendekatan ini dapat meningkatkan kinerja deteksi intrusi dalam hal *scalability, execution time, detection rate,* dan *false alarm*. Model ini dapat mendeteksi semua kelas intrusi yang diuji dengan *accuracy* klasifikasi untuk semua kelas yang diatas 90%. *Accuracy* untuk kelas minoritas (R2L dan U2R) diatas 77%. Kelas Normal dan kelas DoS memiliki *accuracy* diatas 90%, tetapi *accuracy* kelas Probe yang juga sebagai kelas mayoritasberada dibawah 90%.

(Pajouh et al., 2017) mengusulkan model IDS berdasarkan pengklasifikasi dua-tier menggunakan k-Nearest Neighbor (k-NN) dan Naïve Bayes. Mereka menggunakan metoda *linear discriminant analysis* untuk mengurangi dimensi dataset. Pendekatan ini dapat mendeteksi semua kelas serangan. Keakuratan klasifikasinya dari keseluruhan kelas lebih tinggi dari yang dicapai (Bostani & Sheikhan, 2017) Tetapi *accuracy* untuk kelas Probe, R2L, dan U2R lebih rendah.

IDS *hybrid* multi-level menggunakan SVM dan *extreme learning machine* diusulkan (Al-Yaseen et al., 2017). Mereka menggunakan algoritma k-mean yang dimodifikasi untuk meningkatkan kualitas dataset pelatihan. Pendekatan ini juga dapat mendeteksi semua kelas serangan. Akurasi klasifikasi untuk keseluruhan kelas lebih tinggi dari (Pajouh et al., 2017), tetapi akurasi untuk kelas R2L dan U2R lebih rendah.

(Sumaiya Thaseen & Aswani Kumar, 2017) mengusulkan SVM multi-kelas untuk mengenali beragam serangan pada jaringan. Mereka menggunakan

metode *z-score* pada tahap normalisasi dan menggunakan metode pemilihan fitur berbasis *chi-square* untuk memilih atribut yang sesuai. Hasil percobaan menunjukkan bahwa pendekatan mereka dapat mendeteksi semua kelas serangan dengan *accuracy* yang lebih tinggi daripada yang dihasilkan (Al-Yaseen et al., 2017). Mereka menggunakan 31 fitur atau lebih besar dari 50% dari total fitur dataset NSL-KDD.

Tabel 2.1 Perbandingan kinerja model IDS yang membahas aspek *Completeness*

Metode IDS	**∑ fitur**	**Accuracy (%)**					
		Overall	**Normal**	**DoS**	**R2L**	**Probe**	**U2R**
DSSVM (Guo et al., 2014)	52	92.50	98.40	97.21	6.31	87.49	3.07
CANN (Lin et al., 2015)	19	99.46	97.04	99.68	57.05	87.61	3.85
L-SCANN (Ahmad & Muchammad, 2016)	19	95.54	97.50	96.90	88.35	79.43	18.18
CSMN (Muttaqien & Ahmad, 2017)	19	96.87	97.43	97.70	48.57	91.22	9.09
Hybrid SVM and ELM (Al-Yaseen et al., 2017)	---	95.75	98.13	99.54	31.39	87.22	21.93
Naïve Bayes and CF-KNN (Pajouh et al., 2017)	---	94.56	94.56	84.68	34.81	79.76	67.16
Modified OPF (Bostani & Sheikhan, 2017)	---	91.74	98.55	96.89	81.13	85.92	77.98
Chi-FS and multi-class SVM (Sumaiya Thaseen & Aswani Kumar, 2017)	31	98.00	99.60	99.90	98.70	99.20	73.90
MLDR and multi-class SVM (Kumar et al., 2018)	---	98.44	95.74	95.99	78.66	94.97	79.77
OneR-FS and CRF (Mahendiran & Appusamy, 2018)	24	98.15	98.58	98.02	96.11	96.57	92.30

(Mahendiran & Appusamy, 2018) mengusulkan pendekatan lain menggunakan metode pemilihan fitur menggunakan algoritma One-R dan pengklasifikasi berbasis CRF untuk deteksi intrusi. Hasil percobaan mereka menunjukkan bahwa pendekatan ini dapat mendeteksi beragam serangan dengan akurasi tinggi. Akurasi klasifikasi keseluruhan kelas dari pendekatan ini dapat mengungguli kinerja model (Sumaiya Thaseen & Aswani Kumar, 2017), tetapi untuk kelas lain lebih rendah.

(Kumar et al., 2018) juga mengusulkan SVM multi-kelas untuk mendeteksi intrusi. Mereka menggunakan metode pengurangan dimensi multi-

linear (*multi-linear dimensionality reduction* / MLDR) untuk mengurangi dimensi dari dataset. Hasil eksperimen mereka menunjukkan bahwa pendekatan ini juga dapat meningkatkan kinerja SVM dalam akurasi klasifikasi. Kinerja dari pendekatan ini dapat mengungguli kinerja model (Mahendiran & Appusamy, 2018) dalam hal akurasi klasifikasi kelas keseluruhan, tetapi untuk akurasi setiap kelas lebih rendah.

2.2. Kajian Pustaka Dataset NSL-KDD

Dataset NSL-KDD diajukan (Tavallaee et al., 2009) sebagai solusi dari permasalahan yang ada pada *dataset* KDD Cup 1999 (KDD-99). *Dataset* KDD-99 meskipun usianya sudah lebih dari 15 tahun, namun masih umum digunakan dalam penelitian-penelitian sistem deteksi intrusi. Dari studi literatur pada 78 penelitian IDS berbasis pembelajaran mesin *hybrid* dalam rentang waktu tahun 2007-2018 yang kami sajikan pada Tabel 1.1, kami menemukan ada 14 penelitian menggunakan dataset NSL-KDD dan 71 penelitian yang menggunakan dataset KDD Cup 1999 atau variannya.

Tabel 2.2 Daftar Pengelompokan Kelas Serangan pada NSL-KDD

No	Kelas	Nama Tipe Serangan
1	DoS	back, land, neptune, pod, smurf, teardrop
2	R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
3	Probe	ipsweep, nmap, portsweep, satan
4	U2R	buffer_overflow, loadmodule, perl, rootkit

Sumber: (So-In, Mongkonchai, Aimtongkham, Wijitsopon, & Rujirakul, 2014)

Beberapa masalah yang terdapat dalam *dataset* KDD-99 yang sudah ditangani pada NSL-KDD adalah penghapusan data redundan dan proporsi ulang terhadap dataset. NSL-KDD tidak menyertakan data redundan yang ada pada KDD-99 yang dapat mempengaruhi performa dari algoritma pembelajaran. Pada NSL-KDD, dataset KDD-99 telah diproporsi ulang. Dataset ini terbagi menjadi 23 kelas yaitu 1 kelas normal dan 22 kelas tipe serangan. Kelas-kelas serangan tersebut dikelompokkan menjadi 4 kelas (DoS, probe, R2L, dan U2R) seperti

yang disajikan pada Tabel 2.2. Sedangkan deskripsi 41 fitur dari dataset NSL-KDD disajikan pada Tabel 2.3.

Tabel 2.3 Daftar 41 Fitur NSL-KDD

No urut	Nama Fitur	Tipe
1	duration	Integer
2	protocol_type	Nominal
3	service	Nominal
4	flag	Nominal
5	src_bytes	Integer
6	dst_bytes	Integer
7	land	Binary
8	wrong_fragment	Integer
9	urgent	Integer
10	hot	Integer
11	num_failed_logins	Integer
12	logged_in	Binary
13	num_compromised	Integer
14	root_shell	Binary
15	su_attempted	Binary
16	num_root	Integer
17	num_file_creations	Integer
18	num_shells	Integer
19	num_access_files	Integer
20	num_outbound_cmds	Integer
21	is_host_login	Binary
22	is_guest_login	Binary
23	count	Integer
24	srv_count	Integer
25	serror_rate	Float
26	srv_serror_rate	Float
27	rerror_rate	Float
28	srv_rerror_rate	Float
29	same_srv_rate	Float
30	diff_srv_rate	Float
31	srv_diff_host_rate	Float
32	dst_host_count	Float
33	dst_host_srv_count	Float
34	dst_host_same_srv_rate	Float
35	dst_host_diff_srv_rate	Float
36	dst_host_same_src_port_rate	Float
37	dst_host_srv_diff_host_rate	Float
38	dst_host_serror_rate	Float
39	dst_host_srv_serror_rate	Float
40	dst_host_rerror_rate	Float
41	dst_host_srv_rerror_rate	Float
42	class	Nominal

Sumber: (KDD Cup, 1999)

DoS (*Denial of Service*) adalah tipe serangan yang membebani sumber daya komputer (misalnya dengan *synflood* atau *ping of death*) sehingga komputer target mengalami *crash* dan tidak mampu untuk memproses koneksi normal bahkan mengakibatkan user tidak dapat mengakses komputer tersebut.

R2L (r*emote to local*) adalah tipe serangan yang bertujuan untuk mendapatkan akses sebagai pengguna sistem. R2L dilakukan oleh penyerang yang memiliki akses ke sistem dan melakukan eksploitasi untuk mendapatkan akses lokal.

Serangan *Probe* bertujuan untuk mendapatkan informasi tentang status jaringan komputer dengan cara melakukan pemindaian terhadap komputer-komputer dalam jaringan tersebut. Informasi ini dapat digunakan oleh penyerang untuk memetakan jaringan yang berguna dalam melakukan penyerangan berikutnya.

U2R (*user to root*) adalah tipe serangan yang berusaha untuk mendapatkan akses root/admin pada komputer target dengan melakukan eksploitasi celah keamanan sistem. Serangan U2R umumnya dilakukan setelah penyerang mendapatkan akses user normal ke sistem (baik melalui *sniffing*, *social engineering*, ataupun *dictionary attack*).

2.3. Kajian Pustaka Dataset Kyoto2006++

Dataset Kyoto2006++ diajukan oleh (Song et al., 2011) sebagai alternatif untuk *dataset* KDD-99 yang dipandang kurang dapat merefleksikan kondisi jaringan di dunia nyata. Song dkk. membangun sebuah *honeypot* untuk melakukan pengumpulan data pada periode tahun 2006 sampai dengan. 2009. Akses serangan ke *honeypot* tersebut direkam untuk mendapatkan data serangan terhadap jaringan di dunia nyata. Sedangkan data akses normal diperoleh dengan menggunakan mail server dan DNS server yang melakukan akses normal ke *honeypot*. Beragam akses tersebut kemudian digunakan untuk membentuk *dataset* Kyoto2006++. Pelabelan data akses tersebut dilakukan oleh 3 perangkat lunak yaitu Symantec Network Security 7160 (SNS7160), ClamAV, dan Ashula.

Hasil dari penelitian tersebut adalah *dataset* yang memuat tipe-tipe serangan dan fitur-fitur baru yang dapat digunakan untuk penelitian dalam bidang deteksi intrusi pada jaringan. *Dataset* Kyoto2006++ terdiri dari 14 fitur yang diturunkan dari KDD-99 dan 10 fitur tambahan (Tabel 2.4). Informasi kelas dari *dataset* tersimpan pada fitur *label*, yang terdiri dari 3 nilai, yaitu '1' jika data tersebut adalah normal, '-1' jika merupakan data serangan yang dikenali oleh 3 perangkat lunak seperti di atas, dan '-2' jika data merupakan data serangan yang tidak dikenali. *Dataset* terakhir dari Kyoto2006++ adalah data pada bulan Agustus 2009. Dari studi literatur pada 78 penelitian IDS berbasis *hybrid machine learning* dalam rentang waktu tahun 2007-2018 yang kami sajikan pada Tabel 1.1, kami menemukan ada 7 penelitian menggunakan dataset Kyoto 2006++.

Tabel 2.4 Daftar Fitur Kyoto 2006++

No urut	Nama Fitur	Tipe
1	duration	Integer
2	service	Nominal
3	source_bytes	Integer
4	destination_bytes	Integer
5	count	Integer
6	same_srv_rate	Float
7	serror_rate	Float
8	srv_serror_rate	Float
9	dst_host_count	Integer
10	dst_host_srv_count	Integer
11	dst_host_same_src_port_rate	Float
12	dst_host_serror_rate	Float
13	dst_host_srv_serror_rate	Float
14	flag	Nominal
15	ids_detection	Nominal
16	malware_detection	Nominal
17	ashula_detection	Nominal
18	label	Nominal
19	source_IP_address	Nominal
20	source_port_number	Nominal
21	destination_IP_address	Nominal
22	destination_port_number	Nominal
23	start_time	Time
24	duration	Integer

Sumber : (Song et al., 2011)

2.4. Data Normalization pada Penelitian IDS

Data normalization atau *feature scaling* umumnya dilakukan pada tahapan praproses data pada sistem berbasis pembelajaran mesin dan *data mining*. Normalisasi atribut adalah langkah penting ketika berhadapan dengan dataset yang berisi fitur-fitur dengan rentang nilai yang sangat bervariasi. Dalam beberapa algoritma pembelajaran mesin, fungsi objektif tidak akan berfungsi dengan baik tanpa normalisasi akibat rentang nilai data sangat bervariasi. Sebagai contoh, banyak pengklasifikasi menghitung jarak antara dua titik dengan jarak Euclidean. Jika salah satu fitur memiliki rentang nilai yang luas, perhitungan jarak akan didominasi oleh fitur tersebut. Oleh karena itu, rentang semua fitur harus dinormalisasi sehingga setiap fitur berkontribusi secara proporsional terhadap pengukuran jarak.

Untuk aplikasi di IDS, (Wang, Zhang, Gombault, & Knapskog, 2009) membandingkan penerapan empat metode normalisasi, yaitu: Z-score, Min-max, Ordinal, dan Frequency, pada model IDS yang menggunakan algoritma klasifikasi berbasis jarak. Mereka menyimpulkan bahwa Z-score adalah metode dengan kinerja terbaik diikuti oleh Min-Max.

(Li & Liu, 2011) menganalisis dan mensimulasikan enam metode normalisasi, yaitu: zero-mean, sigmoidal, softmax, decimal scaling, max, dan Min-max. Simulasi dilakukan dengan menggunakan pengklasifikasi SVM dan dataset KDD Cup 1999. Hasil uji coba menunjukkan bahwa penggunaan metode normalisasi Min-max menghasilkan *accuracy* yang lebih baik dibanding jika tidak menggunakan normalisasi atau menggunakan metode normalisasi yang lain.

(Said, Stirling, Federolf, & Barker, 2011) membandingkan kinerja metode normalisasi Min-max, Z-score, dan metode Log. Hasil uji coba mereka menunjukkan bahwa metode normalisasi Log menghasilkan kinerja terbaik pada algoritma pengklasifikasi yang menggunakan ukuran jarak Euclidean.

Ketiga penelitian tersebut menghasilkan rekomendasi metode normalisasi yang berbeda yaitu: Min-max, Z-score, dan Log. Oleh karena itu kami akan menggunakan ketiga metode tersebut pada penelitian ini. Selanjutnya kami akan melakukan uji coba untuk mendapatkan metode normalisasi yang dapat

menghindari atau meminimalkan perubahan nilai *mutual information* dari fitur yang diproses karena proses pembulatan hasil normalisasi dengan jumlah digit desimal kecil. Pembahasannya akan kami lakukan pada Bab 4 tentang Pre-processing Data.

Min-max merupakan metode normalisasi yang paling mudah untuk mendapatkan rentang nilai standar 0 hingga 1. Skor dinormalisasi menggunakan persamaan (2.1), dimana x' merupakan skor hasil normalisasi dari skor x, max(X) adalah nilai maksimum, dan min(X) adalah nilai minimum dari skor data yang diproses.

$$x' = (x - min(X))/(max(X) - min(X)) \quad (2.1)$$

Normalisasi Z-score disebut sebagai normalisasi. Skor dinormalisasi menggunakan persamaan (2.2), dimana s' merupakan skor hasil normalisasi dari skor s, dimana σ merupakan standar deviasi dan μ merupakan nilai rata-rata dari sekumpulan skor yang diproses.

$$s' = (s - \mu)/\sigma \quad (2.2)$$

Selanjutnya nilai normalisasi Log dari x dihitung dengan menggunakan persamaan (2.3), dimana x' merupakan nilai hasil normalisasi.

$$x' = log(1 + x) \quad (2.3)$$

2.5. Metode Seleksi Fitur

Seleksi fitur adalah salah satu teknik penting dan banyak digunakan dalam tahapan praproses data untuk IDS (You Chen, Li, Cheng, & Guo, 2006). Kegiatan ini mengurangi jumlah fitur, menghapus data yang tidak relevan, berlebihan, atau berisik, dan membawa efek langsung pada kinerja IDS. Tujuan utama dari seleksi fitur adalah memperoleh kumpulan fitur-fitur terbaik yang dapat meningkatkan kinerja deteksi dari model yang dikembangkan. Secara umum seleksi fitur dikelompokkan menjadi tiga teknik (Frank et al., 2005), yaitu: teknik filter, teknik *wrapper*, dan teknik *hybrid*.

Teknik filter menggunakan pengujian statistik untuk melakukan evaluasi terhadap fitur sehingga teknik ini tidak bergantung kepada algoritma learning

tertentu. Teknik filter akan menghasilkan rangking fitur mulai dari fitur yang paling signifikan sampai yang tidak signifikan. Satu fitur disebut tidak signifikan jika fitur tersebut tidak memiliki pengaruh dalam penentuan label klasifikasi.

Teknik *wrapper* melakukan seleksi fitur dengan membentuk subset-subset yang terdiri dari kombinasi yang mungkin dari fitur dataset. Kemudian masing-masing subset tersebut akan dievaluasi dengan algoritma pengklasifikasi untuk mengetahui kinerja deteksi yang dihasilkan subset data tersebut. Teknik wrapper dapat memberikan hasil yang lebih baik daripada teknik filter, namun membutuhkan waktu yang lama dan tingkat komputasi yang lebih tinggi. Metode *hybrid* menggabungkan pendekatan filter dan *wrapper*.

Pada penelitian ini kami akan mengajukan metode seleksi fitur *hybrid*. Kami menggunakan teknik filter yang menggunakan *information-gain* untuk evaluasi fitur dan algoritma *Ranker* untuk mengurutkan fitur berdasarkan nilai *information-gain* dari fitur. Hasil urutan fitur tersebut selanjutnya kami proses menggunakan teknik *wrapper* menggunakan pengklasifikasi *support vector machine* (SVM). Subset yang diuji dibentuk dari n fitur urutan teratas, dimana nilai n dari 2 sampai dengan jumlah fitur dari dataset.

Untuk mendapatkan subset fitur yang mendukung peningkatan kinerja *minority-class* dan tetap mempertahankan kinerja kelas lainnya tetap tinggi, kami melakukan dua pendekatan pada tahapan teknik filter. Pendekatan pertama, nilai *information-gain* dari fitur dikalikan dengan bobot kelas. Pendekatan kedua, dilakukan dengan menggunakan subset data yang hanya terdiri dari kelas aktivitas normal dan *minority-class* dalam proses penyusunan rangking fitur. Kedua pendekatan ini diharapkan dapat meningkatkan rangking dari fitur-fitur yang berpengaruh dalam pendeteksian jenis serangan yang tergolong *minority-class*. Sedangkan pada tahapan teknik *wrapper*, pemilihan nilai n optimal dievaluasi menggunakan nilai CPI yang merupakan indek kinerja gabungan yang merepresentasikan kondisi pencapaian overall *accuracy* yang tinggi dengan jumlah *false negative* yang rendah, dan *sensitivity* yang tinggi pada *minority class*. Pembahasan lebih rinci tentang metode seleksi fitur yang kami usulkan pada penelitian ini akan dibahas pada bagian 3.4 dan 4.2.

2.6. Satuan Ukur Kinerja pada IDS

Dari studi literatur pada 78 makalah penelitian IDS berbasis pembelajaran mesin *hybrid* dalam rentang waktu tahun 2007-2018 (Setiawan, Djanali, & Ahmad, 2017) diketahui bahwa satuan ukur kinerja yang banyak digunakan oleh peneliti IDS adalah *accuracy* (ACC), *detection rate* (DR), *false positive* (FP), *false negative* (FN), true positive (TP), dan *false alarm* (FA). Satuan ukur kinerja lainnya yang digunakan adalah *sensitivity*, *specificity*, *F-measure*, *precision*, *receiver operating characteristic*, dan *run-time*.

2.7. Composite Indicators

Indikator komposit telah diakui sebagai alat yang berguna untuk pembandingan, analisis kebijakan, dan pemantauan kinerja dalam berbagai bidang, seperti di bidang lingkungan, ekonomi, informasi, inovasi, dan pengetahuan (Nardo & Saisana, 2008; Peng, Wu, Fu, & Lai, 2017; Zhou, Ang, & Poh, 2007). Indikator komposit ideal untuk mengukur konsep multi dimensi yang tidak dapat dilakukan oleh indikator individu. Berikut definisi indikator komposit yang ditafsirkan dari *OECD Glossary of Statistics Requirements*, ‘‘*A composite indicator is constructed when individual indicators are compiled into a single index, by an underlying model of the multi-dimensional concept that is being measured.*’’ Dimana indikator gabungan dibuat dengan melakukan kompilasi beberapa indikator individual menjadi indeks tunggal, yang didasari konsep multidimensi yang sedang diukur.

Pada penelitian ini *composite performance index* (CPI) dibangun dari empat ukuran kinerja tunggal yaitu *accuracy* untuk keseluruhan kelas, *false-negative* dari semua serangan, *sensitivity* dari kelas serangan yang memiliki jumlah data training paling sedikit, dan *sensitivity* dari kelas serangan yang memiliki jumlah data training paling sedikit kedua). Indeks ini digunakan untuk mendapatkan kondisi kinerja dimana model memiliki *accuracy* yang tinggi dan semua kelas terdeteksi. Kami menerapkan indeks ini dalam pemilihan fitur dan optimalisasi bobot kelas minoritas di WSVM.

Kami menerapkan pendekatan *multiple attribute decision making* (MADM) untuk membangun CPI. Kami mendapat inspirasi dari *Simple Multi-Attribute Rating Technique*, salah satu model MADM yang dibahas pada (Ishizaka & Siraj, 2018) untuk membangun indek ini. Kami juga mengimplementasikan pendekatan subyektif (Dong, Liu, Liang, Chiclana, & Herrera-Viedma, 2018) untuk mendapatkan bobot dari atribut-atribut. Sehingga nantinya persyaratan penentuan bobot atribut sesuai pilihan dari pembuat keputusan. Karena kami hanya memerlukan peringkat dari atribut, maka kami menggunakan tingkatan prioritas atribut untuk menentukan bobot dari atribut.

Untuk pembentukan agregasi, kami mendefinisikan satu set bobot non-negatif w_j untuk merepresentasikan kontribusi atribut atau indikator I_j terhadap nilai CPI. Indikator-indikator disusun berdasar prioritas dalam urutan menurun sedemikian rupa sehingga $w_1 \geq w_2 \geq \cdots \geq w_J$, dimana J adalah jumlah indikator dan j adalah nomor urut indikator. Skor dari CPI dinyatakan sebagai jumlah ukuran kinerja tertimbang dari beberapa indikator. Model linear dari pembobotan untuk keperluan agregasi dari CPI disajikan pada persamaan (2.4). Persamaan (2.5) - (2.8) merupakan batasan-batasan dalam penentuan nilai bobot dari indikator. Persamaan (2.5) dan (2.6) merupakan batasan normalisasi untuk nilai bobot sehingga nilainya selalu dalam kisaran dari 0 sampai 1 dan jumlah total semua bobot sama dengan 1. Selain itu persamaan (2.7) digunakan untuk memastikan urutan peringkat indikator. Persamaan (2.8) digunakan untuk menghitung nilai bobot indikator berdasar urutan prioritas indikator (Setiawan, Djanali, & Ahmad, 2019), yang memenuhi batasan-batasan pada persamaan (2.5) - (2.7).

$$CPI = \sum_{j=1}^{J} w_j . I_j \tag{2.4}$$

dimana

$$\sum_{j=1}^{J} w_j = 1, \tag{2.5}$$

$$w_j > 0, \quad j = 1,2, \dots, J \tag{2.6}$$

$$w_j - w_{(j+1)} \geq 0, \quad j = 1, 2, \dots, (J-1) \tag{2.7}$$

$$w_j = \frac{(J - j + 1)}{\sum_{n=1}^{J} n}, \quad j = 1,2, \dots, J \tag{2.8}$$

Kami menggunakan *accuracy* sebagai indikator prioritas pertama, *false-negative* sebagai indikator prioritas kedua, dan *sensitivity* dari *minority classes* U2R dan R2L sebagai prioritas ketiga dan keempat. Selanjutnya CPI dihitung menggunakan persamaan (2.9), dimana w_1 merupakan bobot dari *accuracy*, w_2 merupakan bobot dari *false negative*, w_3 merupakan bobot dari *minority class* pertama yaitu kelas U2R, dan w_4 merupakan bobot dari *minority class* kedua yaitu kelas R2L.

$$\begin{aligned} CPI = \quad & w_1 \times Accuracy_{\,Overall} + \; w_2 \times FalseNegative_{\,Overall} \\ & + \; w_3 \times Sensitivity_{U2R} \; + \; w_4 \times Sensitivity_{R2L} \end{aligned} \tag{2.9}$$

Selanjutnya kami menghitung nilai dari w_1, w_2, w_3, dan w_4 menggunakan persamaan (2.13) dengan $J = 4$ atau sejumlah indikator yang digunakan.

$$w_1 = \frac{(4 - 1 + 1)}{1 + 2 + 3 + 4} = \frac{4}{10} = 0.4$$

Dari perhitungan itu, kita akan mendapatkan nilai w_1 = 0.4, w_2 = 0.3, w_3 = 0.2, dan w_4 = 0.1. Sehingga persamaan CPI akan menjadi seperti yang ditunjukkan pada persamaan (2.10).

$$\begin{aligned} CPI = \quad & 0.4 \times Accuracy_{\,Overall} + \; 0.3 \times FalseNegative_{\,Overall} \\ & + 0.2 \times Sensitivity_{U2R} \; + \; 0.1 \times Sensitivity_{R2L} \end{aligned} \tag{2.10}$$

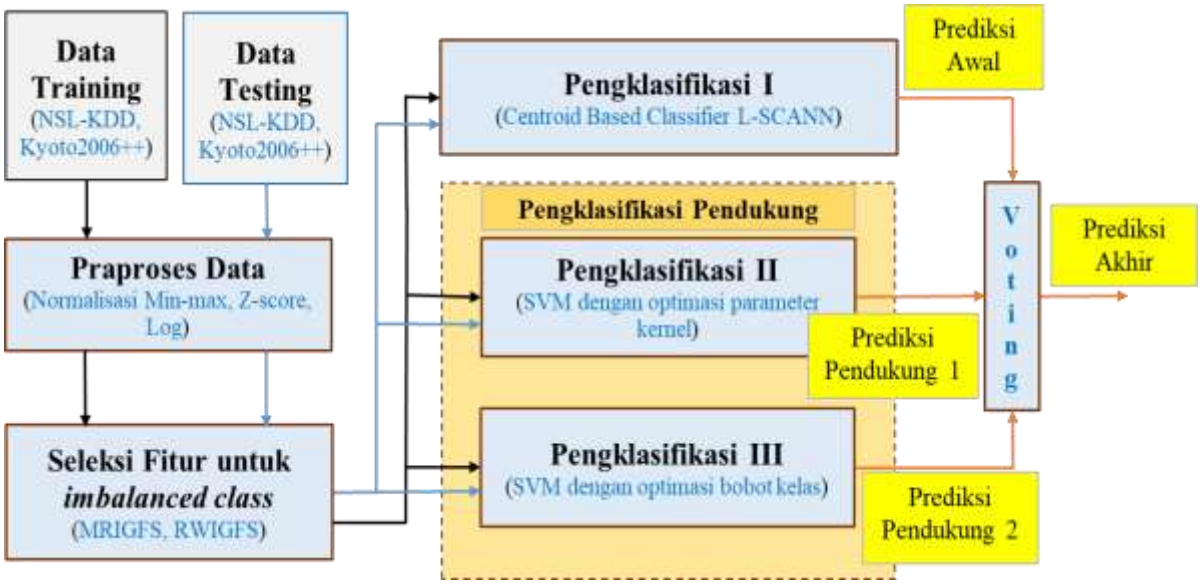
Halaman ini sengaja dikosongkan

BAB 3
METODE PENELITIAN

Metode *Centroid-based Classification* (CBC) merupakan salah satu bentuk pendekatan pembelajaran mesin *hybrid* yang spesifik, yang mengkombinasikan algoritma pengklasteran dan algoritma pengklasifikasi (Lin et al., 2015). Algoritma pengklasteran sebagai komponen pertama akan memproses dataset input menjadi fitur representatif baru. Selanjutnya, pengklasifikasi sebagai komponen kedua akan memproses fitur representatif tersebut untuk menghasilkan hasil akhir berupa prediksi. Model ini dianggap sangat efisien dalam tahap pelatihan dan klasifikasi, karena proses perhitungan hanya didasarkan pada pusat-pusat klaster (*centroid)* dan tidak melibatkan seluruh data pelatihan sehingga waktu pemrosesan menjadi lebih pendek (Lin et al., 2015). Penelitian terdahulu di bidang IDS yang menggunakan metode CBC: (Ahmad & Muchammad, 2016; Guo et al., 2014; Lin et al., 2015; Muttaqien & Ahmad, 2017) menunjukkan bahwa IDS berbasis CBC mempunyai kinerja deteksi yang tinggi tetapi masih kesulitan dalam mendeteksi serangan jenis serangan, terutama jenis serangan U2R dan R2L yang merupakan *minority class* pada dataset KDD-Cup 1999 dan NSL-KDD.

Pada penelitian ini, kami berupaya meningkatkan kinerja IDS berbasis CBC dengan menggabungkan beberapa pendekatan. Pada tahapan pra-proses data, kami berusaha menghindari adanya perubahan nilai mutual information dari fitur-fitur dataset dengan melakukan evaluasi terhadap metode normalisasi yang digunakan. Selanjutnya dilakukan seleksi fitur untuk mendapatkan fitur-fitur yang mendukung pendeteksian serangan. Dua metode seleksi fitur untuk kondisi *imbalanced class*, yaitu MRIGFS dan RWIGFS, diajukan untuk mendapatkan fitur-fitur yang lebih mendukung deteksi terhadap *minority class*. Pada proses klasifikasi pada CBC yang menggunakan algoritma k-NN dilakukan optimasi parameter k. Kemudian setelah klasifikasi dengan CBC dilakukan proses validasi menggunakan dua pengklasifikasi berbasis SVM, yaitu: SVM dengan optimasi parameter kernel RBF (SVM-OP) dan SVM dengan optimasi bobot kelas (SVM-OW). Gambar 3.1

menunjukkan bagan IDS dengan pendekatan *ensemble-voting* tiga pengklasifikasi yang diajukan pada penelitian ini.



Gambar 3.1 Blok diagram sistem deteksi intrusi berbasis *ensemble-voting* tiga pengklasifikasi

Tahapan penelitian yang dilakukan adalah sebagai berikut:

1. Praproses data. Pada tahapan ini dilakukan konversi nilai fitur nominal ke numerik dan penyeragaman rentang nilai dari fitur-fitur dengan menggunakan metode normalisasi. Tiga metode normalisasi digunakan dan dibandingkan pada penelitian ini, yaitu: Min-max, Zscore, dan Log.
2. Seleksi fitur. Proses penyeleksian fitur dilakukan menggunakan metode seleksi fitur untuk *imbalanced-class*. Pada penelitian ini kami mengajukan dua metode seleksi fitur yaitu MRIGFS dan RWIGFS.
3. Klasifikasi pertama menggunakan pengklasifikasi I. Pengklasifikasi I merupakan pengklasifikasi utama yang berbasis CBC. Pada penelitian ini kami menggunakan L-SCANN (Muttaqien & Ahmad, 2017) untuk menguji pendekatan yang kami usulkan.
4. Klasifikasi kedua dilakukan dengan menggunakan pengklasifikasi SVM-OP (pengklasifikasi II). SVM-OP diimplementasikan menggunakan modul

Lib-SVM pada aplikasi data mining WEKA versi 3.8.3 (Waikato University, 2018).

5. Klasifikasi ketiga dilakukan dengan menggunakan pengklasifikasi SVM-OW (pengklasifikasi III). SVM-OW juga diimplementasikan menggunakan modul Lib-SVM pada aplikasi data mining WEKA versi 3.8.3 (Waikato University, 2018).
6. Proses *Voting*. Proses ini merepresentasikan proses validasi output dari model IDS berbasis CBC dengan menggunakan dua pengklasifikasi SVM-OP dan SVM-OW dengan teknik *ensemble-voting*.
7. Melakukan uji model dan pengukuran kinerja.

Pada bagian selanjutnya kami membahas secara lebih rinci dari tiap tahapan proses dilakukan. Pembahasan secara berurutan mulai dari dataset IDS yang digunakan pada penelitian ini, proses konversi fitur nominal ke numerik, proses normalisasi, pemodelan IDS berbasis CBC, pemodelan IDS berbasis SVM-OP, pemodelan IDS berbasis SVM-OW, pemodelan IDS berbasis *ensemble-voting* tiga pengklasifikasi, dan terakhir membahas satuan ukur yang digunakan untuk mengevaluasi kinerja model IDS.

3.1. Dataset IDS

Dataset yang digunakan dalam penelitian ini adalah dataset NSL-KDD dan Kyoto2006++. Kedua dataset ini tersedia untuk publik dan banyak digunakan oleh para peneliti pada topik sistem deteksi intrusi pada jaringan. Dataset NSL-KDD yang digunakan dalam penelitian ini adalah file dataset KDD-Train+.csv yang diperoleh dari laman (*http://nsl.cs.unb.ca/NSL-KDD/*). Dataset ini kemudian dikonversi ke bentuk *5-class problem* sesuai dengan kelompok serangan seperti pada Tabel 3.1.

Dataset Kyoto2006++ yang digunakan dalam penelitian ini adalah dataset 20090730.txt yang merupakan rekaman akses ke *honeypot* pada tanggal 30 Juli 2009 yang diperoleh dari laman (*http://www.takakura.com/Kyoto_data)*. Fitur yang

digunakan pada penelitian ini adalah fitur no 1 s.d. 14 pada Tabel 2.4 dan 1 fitur *label*, sedangkan fitur sisanya tidak digunakan. Dataset ini kemudian dikonversi ke bentuk *2-class problem* seperti pada Tabel 3.1.

Tabel 3.1 Konversi Kelas pada Dataset NSL-KDD dan Dataset Kyoto2006++

No	Dataset	Kelas Sebelum Konversi	Kelas Setelah Konversi	Jumlah Record
1	NSL-KDD	normal	Normal	67343
		back, land, neptune, pod, smurf, teardrop	DoS	45927
		buffer_overflow, loadmodule, perl, rootkit	U2R	130
		ipsweep, nmap, portsweep, satan	Probe	11306
		ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster	R2L	1990
2	Kyoto2006++	1	Normal	63263
		-1, -2	Attack	5761

3.2. Konversi Fitur Nominal ke Numerik

Pada penelitian ini kami menggunakan metode *arbitrary* (Hernández-Pereira, Suárez-Romero, Fontenla-Romero, & Alonso-Betanzos, 2009) untuk melakukan konversi nilai nominal dari dataset ke nilai numerik. Metode ini membangun korespondensi antara setiap kategori fitur simbolik dengan nilai integer yang berurutan. Nilai nominal dipetakan ke dalam nilai integer mulai dari 0 hingga $C - 1$, dimana C adalah jumlah kategori yang ada pada fitur nominal tersebut.

Tabel 3.2 Konversi untuk fitur "protocol_type"

Nominal	Numerik
icmp	0
tcp	1
udp	2

Tabel 3.3 Konversi untuk fitur "flag"

Nominal	Numerik	Nominal	Numerik
OTH	0	S0	5
REJ	1	S1	6
RSTO	2	S2	7
RSTOS0	3	S3	8
RSTR	4	SF	9
		SH	10

Tabel 3.4 Konversi untuk fitur "service"

Nominal	Numerik	Nominal	Numerik	Nominal	Numerik	Nominal	Numerik
aol	0	gopher	18	netbios_ns	36	sql_net	54
auth	1	harvest	19	netbios_ssn	37	ssh	55
bgp	2	hostnames	20	netstat	38	sunrpc	56
courier	3	http	21	nnsp	39	supdup	57
csnet_ns	4	http_2784	22	nntp	40	systat	58
ctf	5	http_443	23	ntp_u	41	telnet	59
daytime	6	http_8001	24	other	42	tftp_u	60
discard	7	imap4	25	pm_dump	43	tim_i	61
domain	8	IRC	26	pop_2	44	time	62
domain_u	9	iso_tsap	27	pop_3	45	urh_i	63
echo	10	klogin	28	printer	46	urp_i	64
eco_i	11	kshell	29	private	47	uucp	65
ecr_i	12	ldap	30	r2l4	48	uucp_path	66
efs	13	link	31	red_i	49	vmnet	67
exec	14	login	32	remote_job	50	whois	68
finger	15	mtp	33	rje	51	X11	69
ftp	16	name	34	shell	52	Z39_50	70
ftp_data	17	netbios_dgm	35	smtp	53		

3.3. Proses Normalisasi

Pada penelitian ini kami membandingkan penggunaan tiga metode normalisasi, yaitu Min-max, Z-score, dan Log; untuk mendapatkan metode normalisasi yang dapat meningkatkan kecepatan deteksi, meningkatkan ketepatan deteksi, serta menghindari perubahan nilai *mutual information* dari fitur yang diproses.

Untuk mengetahui pengaruh penggunaan metode normalisasi dan jumlah digit desimal yang digunakan dalam proses pembulatan hasil normalisasi terhadap nilai *mutual informatio*n fitur yang diproses kami menggunakan satuan ukur

information-gain. Kami membandingkan nilai *information-gain* dari fitur-fitur sebelum normalisasi dan sesudah pembulatan hasil normalisasi. *Information-gain* merupakan salah satu standar untuk ukuran kualitas atribut. Dalam penelitian ini, kami juga menggunakan *information-gain* sebagai satuan ukur evaluasi atribut pada metode seleksi fitur yang kami usulkan. Pengukuran berdasarkan kandungan informasi menggunakan *information-gain* banyak digunakan dalam pembelajaran mesin. Jumlah informasi dari hasil X_j didefinisikan sebagai logaritmik negatif dari probabilitasnya, formula perhitungannya ditunjukkan pada persamaan (3.1).

$$I(X_j) = -log_2 P(X_j) \tag{3.1}$$

Jumlah rata-rata informasi disebut sebagai *entropy* dari hasil. Jika hasil dari suatu percobaan X_j memiliki m kemungkinan dimana $j = 1..m$ dan $\sum_j P(X_j) = 1$, maka *entropy* dari hasil dapat didefinisikan sebagai:

$$H(X) = -\sum_{j}^{m} P(X_j)\, log_2 P(X_j) \tag{3.2}$$

Information-gain juga dikenal sebagai *mutual information* dari fitur dengan label kelas, yang nilainya ditentukan sebagai jumlah informasi yang dihasilkan dari atribut yang digunakan untuk menentukan kelas. Perhitungannya seperti yang ditunjukkan pada persamaan (3.3) dan (3.4).

$$InfoGain(A) = H_C - H_{C|A} \tag{3.3}$$

$$H_C - H_{C|A} = H_C + H_A - H_{CA} = I(A;C) = H_A - H_{A|C} = I(C;A) \tag{3.4}$$

Untuk mengetahui pengaruh penggunaan metode normalisasi dan jumlah digit desimal yang digunakan dalam proses pembulatan hasil normalisasi terhadap kinerja pengklasifikasi, kami melakukan uji coba menggunakan dataset pelatihan NSL-KDD 100% dan tiga pengklasifikasi yaitu SVM dan k-NN pada aplikasi data mining WEKA versi 3.8.3(Waikato University, 2018) dan pada L-SCANN (Muttaqien & Ahmad, 2017). Kami melakukan proses pembulatan hasil normalisasi menggunakan 2 digit sampai dengan 6 digit tempat desimal. Untuk menghindari

bias akibat proses konversi nilai fitur nominal ke numerik, uji coba hanya dilakukan pada semua fitur *continuous* (numerik).

Analisis pengaruh penggunaan metode normalisasi dan jumlah digit yang digunakan dalam proses pembulatan hasil normalisasi terhadap hasil seleksi fitur juga dilakukan. Kami menggunakan metode seleksi fitur Rank+InformationGain yang menggunakan algoritma Ranker untuk pencarian dan *information-gain* sebagai *attribute evaluation*.

3.4. Proses Seleksi Fitur

Pada penelitian ini kami mengajukan dua metode seleksi fitur untuk *imbalanced-class*, yaitu m-RIGFS dan RWIGFS. Kedua metode ini menggunakan dua tahapan proses. Tahapan pertama menggunakan metode filter berbasis *information-gain* dan algoritma *ranker* untuk membuat peringkat fitur berdasar nilai *information-gain* yang dimiliki fitur. Secara umum algoritma dari seleksi fitur tahapan pertama yang diajukan dapat dilihat pada Gambar 3.1.

Input:
$D(F_0, F_1, \dots, F_{n-1})$ // Dataset training dengan N fitur
δ // Nilai batas untuk berhenti
$AttributeVal[0 \dots n-1]$ // nilai evaluasi dari masing-
// masing atribut
Output: // Subset berdasar urutan nilai
S_{Rank} // evaluasi atribut
// dari besar ke kecil

```
01  begin
02      initialize: δ = N
03      n = 0;
04      do begin                                  // evaluasi atribut dengan metode
05      γ_n = eval(F_n, D, M);                    // evaluasi M
06      AttributeVal[n] = γ_n;                    // dari atribut ke-0 sampai ke-(n-1)
07      n = n + 1;
08      end until (n = δ)
09      S_Rank = Ranker(D, AttributeVal[ ]);      //urutkan fitur berdasarkan nilai
                                                  //evaluasi fitur
10      return S_Rank

11  end;
```

Gambar 3.2 Algoritma seleksi fitur dengan teknik filter

Tahapan kedua untuk mendapatkan jumlah top fitur optimal digunakan metode *wrapper* dengan pengklasifikasi SVM dari modul Lib-SVM pada aplikasi data mining WEKA versi 3.8.3(Waikato University, 2018). Pada penelitian ini kami melakukan uji coba dari 2 sampai dengan 32 top fitur. Pemilihan jumlah fitur optimal dievaluasi menggunakan nilai CPI yang formulanya dapat dilihat pada persamaan (2.10).

Pada penelitian ini kami akan mengajukan dua metode seleksi fitur yang mendukung kondisi imbalanced-class. Kedua metode tersebut kami berinama metode *modified rank information-gain feature selection* (MRIGFS) dan *rank weight information-gain feature selection* (RWIGFS).

Metode *rank information-gain feature selection* (RIGFS) merupakan metode seleksi fitur dengan evaluasi fitur menggunakan nilai *information-gain* dari fitur terhadap kelas klasifikasi dan hasilnya diurutkan dari besar ke kecil dengan menggunakan algoritma *Ranker*. Perhitungan dari *information-gain* dapat dilihat pada persamaan (3.1) – (3.4).

Metode *rank weighting information gain feature selection* (RWIGFS) mengadopsi pendekatan cost sensitive learning dengan melakukan pembobotan pada perhitungan nilai Information-Gain (*Weighting Information-Gain*) dari suatu fitur. *Weighting Information-Gain* dari suatu fitur didapatkan dengan menjumlahkan nilai Information-Gain fitur terhadap suatu kelas yang telah dikalikan dengan bobot klas. Dimana bobot dari suatu kelas berbanding terbalik dengan jumlah data pada kelas tersebut, sehingga bobot dari suatu kelas sama dengan (1 / jumlah sampel dalam kelas).

Berikut adalah tahapan proses pada RWIGFS:

Langkah ke-1: Untuk dataset dengan n-class (C1..Cn), terlebih dahulu dibentuk n dataset dengan 2 class.

Misal untuk dataset dengan 3 class:

Dataset ke-1, memiliki class C1 dan C-sisa (yang merupakan gabungan dari C2 dan C3)

Dataset ke-2, memiliki class C2 dan C-sisa (yang merupakan gabungan dari C1 dan C3)

Dataset ke-3, memiliki class C3 dan C-sisa (yang merupakan gabungan dari C1 dan C2)

Langkah ke-2: menghitung bobot masing-masing class yang nilainya adalah jumlah data sampel pada masing-masing class dibagi total data sampel.

Langkah ke-3: menghitung nilai *information-gain* fitur pada dataset ke-1, ke-2, dan ke-3.

Langkah ke-4: menggabungkan nilai *information-gain* dari masing-masing fitur yang didapat dari ketiga dataset setelah dikalikan dengan bobot kelas.

Total IG (Fitur ke-n) =

(IG fitur ke-n dari dataset ke-1 x bobot kelas C1) +
(IG fitur ke-n dari dataset ke-2 x bobot kelas C2) +
(IG fitur ke-n dari dataset ke-3 x bobot kelas C3)

Langkah ke-5: membuat ranking fitur berdasarkan nilai total *information-gain* dari fitur.

Langkah ke-6: melakukan proses *wrapper* dengan pengklasifikasi SVM standar.

Metode MRIGFS ini mengadopsi pendekatan *cost sensitive learning* secara tidak langsung. Pemilihan fitur dilakukan dengan menggunakan metode RIGFS yang dimodifikasi. Tujuan modifikasi adalah untuk mendapatkan subset fitur yang mendukung dalam mendeteksi kelas minoritas. Pemilihan fitur dilakukan pada dataset sementara yang hanya terdiri dari kelas Normal dan 50% dari kelas serangan yang dianggap sebagai kelas minoritas, yaitu kelas R2L dan kelas U2R. Selanjutnya, subset fitur digunakan untuk menghasilkan dataset baru dari dataset lengkap yang terdiri dari semua kelas.

Berikut adalah tahapan proses pada MRIGFS:

Langkah ke-1: membentuk dataset temporari yang hanya terdiri dari class dengan jumlah data sampel tertinggi (*majority class*) dengan class yang memiliki jumlah data sampel terendah (*minority class*). Pada penelitian ini yang menggunakan dataset NSL-KDD yang terdiri dari 5 kelas serangan, dengan urutan kelas berdasarkan jumlah

sampel adalah Normal-DoS-Probe-R2L-U2R, dataset temporari dibentuk dari kelas Normal, R2L, dan U2R.

Langkah ke-2: menghitung nilai *information-gain* fitur pada dataset temporari dan membuat rangking fitur.

Langkah ke-3: melakukan proses *wrapper* dengan algoritma SVM standar pada dataset yang asli berdasar rangking fitur yang didapat dari langkah ke-2.

3.5. Pemodelan IDS berbasis CBC

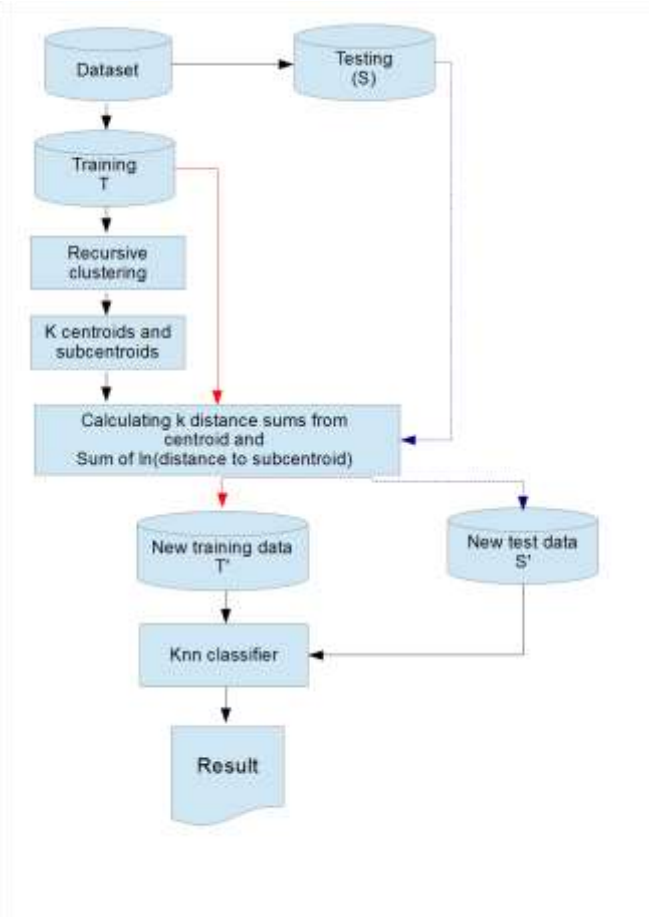
Pada penelitian ini kami menggunakan *Logarithmic subcentroid and nearest neighbor* (L-SCANN), IDS berbasis CBC yang dibahas pada penelitian (Muchammad & Ahmad, 2015), (Ahmad & Muchammad, 2016), dan (Muttaqien & Ahmad, 2017). Metode ini memiliki dua tahapan proses yaitu tahap pembangkitan fitur dan tahap klasifikasi. Alur sistem L-SCANN disajikan pada Gambar 3.2.

Pada tahapan pertama yang merupakan tahapan pembangkitan fitur, dilakukan proses pengklasteran terhadap dataset pelatihan T dengan menggunakan metode *recursive clustering*. Ada dua nilai ambang batas yang terlebih dahulu ditetapkan oleh pengguna di awal proses. Yang pertama adalah nilai ambang batas U untuk proses pembentukan klaster, dan yang kedua adalah nilai ambang batas O yang digunakan pada proses pembentukan sub klaster. Dari hasil ekperimen pada (Muchammad & Ahmad, 2015) didapatkan nilai optimal untuk U dan O untuk dataset NSL-KDD adalah 0.2 dan 4, sedangkan untuk dataset Kyoto2006++ adalah 0.1 dan 6.

Langkah *recursive clustering* dilakukan dengan membentuk data menjadi 2 klaster menggunakan algoritma K-*means* dengan k=2. Proses ini digunakan untuk membentuk klaster dan mencari pusat klaster (*centroid*). Jika klaster yang dihasilkan memiliki *gini impurity index* atau *entropy* melebihi ambang batas U, maka klaster tersebut dipecah lagi menjadi 2 klaster. Hal ini dilakukan sampai setiap klaster memiliki *gini impurity index* di bawah ambang batas U. Perhitungan

gini impurity index ditunjukkan pada persamaan (3.5), dimana m adalah jumlah label pada dataset dan f_i adalah frekuensi label ke-i. Pada penerapan di IDS ditetapkan nilai m=2, yang merepresentasikan kondisi aktivitas normal dan aktivitas yang dianggap serangan.

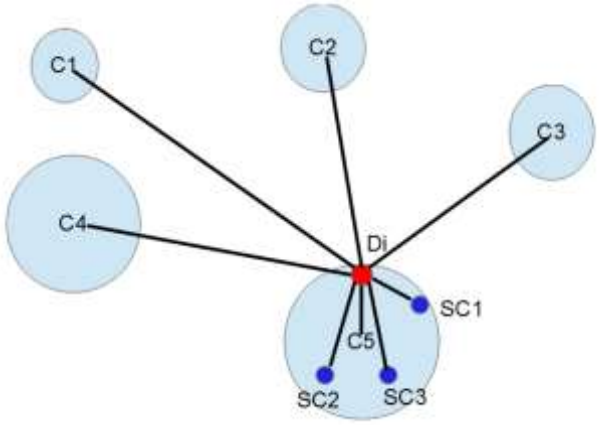
$$GI = \sum_{i=1}^{m} f_i(1 - f_i) \tag{3.5}$$



Gambar 3.3 Alur Sistem IDS dengan L-SCANN (Muchammad & Ahmad, 2015)

Proses berikutnya yang dilakukan adalah mencari *subcentroid* di tiap klaster. *Subcentroid* didapatkan dengan melakukan pengklasteran menggunakan algoritma k-*means* pada tiap *cluster* dengan nilai k=O jika jumlah anggota klaster lebih dari O, jika tidak maka ditetapkan nilai k=1.

Setelah *cluster*, *centroid* dan *subcentroid* didapatkan, proses berikutnya memasukkan tiap data $D_i \in T$ ke klaster yang jarak *centroid*-nya yang paling dekat. Gambar 3.3 menunjukkan jarak apa saja yang diperlukan untuk membangkitkan fitur. Pada gambar tersebut dianggap terdapat 5 *centroid* dan data D_i masuk ke klaster 5 karena jaraknya lebih dekat ke *centroid* C5. Fitur representatif $D_{i'}$ dibentuk dengan menjumlahkan jarak dari D_i ke tiap *centroid* (C1, C2, C3, C4, C5) dan logaritma natural dari D_i ke tiap *subcentroid* di klasternya (SC1, SC2, SC3).



Gambar 3.4 Jarak yang digunakan untuk proses pembangkitan fitur pada L-SCANN (Muchammad & Ahmad, 2015)

Proses pembangkitan fitur untuk data pengujian S dilakukan dengan memasukkan data pengujian ke klaster yang jarak ke *centroid*-nya paling dekat. Langkah berikutnya sama dengan pembangkitan fitur pada data pelatihan yaitu dengan mencari jarak dari data ke tiap pusat klaster (*centroid*) dan logaritma natural dari jarak data ke *subcentroid*. Keluaran dari tahapan ini adalah T' dan S' yaitu data pelatihan dan data pengujian yang telah ditransformasikan. Fitur representatif yang dibangkitkan pada langkah ini berupa fitur 1 dimensi.

Tahapan kedua yang merupakan tahapan klasifikasi, dilakukan dengan menggunakan pengklasifikasi k-*nearest neighbor* (k-NN). Untuk mengklasifikasi data pengujian, pengklasifikasi k-NN hanya menggunakan data pelatihan berada pada klaster yang sama dengan data yang diuji, tidak menggunakan semua data pelatihan T'. Pada penelitian ini kami melakukan optimasi parameter k dari k-NN.

Optimasi dilakukan dengan menggunakan nilai k=1, 3, 5, 7, 9, 11, 13, 15, 17, 19. Pada implementasi dengan dataset NSL-KDD digunakan nilai parameter U=0.2 dan O=4, sedangkan pada dataset Kyoto2006++ digunakan nilai parameter U=0.1 dan O=0.6. Uji coba pelatihan dan pengujian dilakukan dengan metoda 10-*fold cross validation*, dimana dataset dibagi menjadi sepuluh subset yang tidak terduplikasi, dan sembilan dari sepuluh subset digunakan untuk pelatihan dan sisanya untuk pengujian. Dengan demikian, pengklasifikasi akan dilatih dan diuji 10 kali. Hasil pengujian dievaluasi menggunakan nilai CPI yang formulanya dapat dilihat pada persamaan (2.10). Nilai k optimal direpresentasikan dengan nilai CPI tertinggi yang dicapai dalam pengujian.

3.6. Pemodelan IDS berbasis SVM-OP

Support vector machines (SVM) adalah algoritma *supervised learning* yang dikenalkan oleh Boser, Guyon, dan Vapnik (Boser, Guyon, & Vapnik, 1992). SVM melakukan klasifikasi data dengan cara mencari *hyperplane* dengan margin terbesar. Dengan N data pelatihan $\{x_1, y_1\}, \dots, \{x_N, y_N\}$, dimana $x_i \in R^m$ adalah vektor fitur m-dimensi yang merepresentasikan data pelatihan ke- i, dan $y_i \in \{-1,1\}$ merupakan label kelas dari x_i. *Hyperplane* digambarkan sebagai persamaan (3.6), dimana $w \in R^m$ dan b adalah skalar.

$$g(x) = w^T x + b \tag{3.6}$$

Ketika penerapan SVM linear tidak menghasilkan kinerja yang memuaskan, disarankan untuk menggunakan SVM nonlinear. Ide dasarnya adalah menggunakan pemetaan nonlinear $\emptyset(x)$ untuk memetakan x ke ruang dimensi yang lebih tinggi dimana *hyperplane* optimal ditemukan. Pemetaan linear dilakukan dengan menggunakan fungsi kernel $K(x_i, x_j)$ yang menghitung *inner product* dari vektor $\emptyset(x_i)$ dan $\emptyset(x_j)$. Fungsi kernel yang umum digunakan adalah *polynomial function* yang direpresentasikan persamaan (3.7) dan *radial basis function* (RBF) yang direpresentasikan persamaan (3.8). Pada penelitian ini kami menggunakan kernel RBF.

$$K(x_i, x_j) = (x_i^T x_j + 1)^d \quad (3.7)$$

$$K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right) \quad (3.8)$$

Pada tahap klasifikasi, label kelas y_{SVM} dari sampel x ditentukan dengan tanda fungsi keputusan pada persamaan (3.9), dimana α_i merupakan koefisien Lagrange multiplier untuk sampel ke-i.

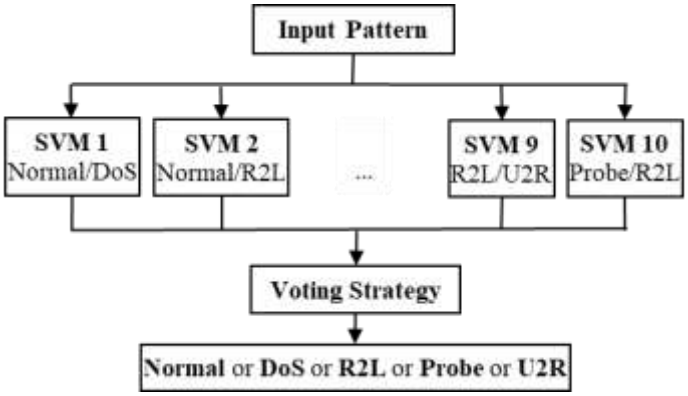
$$f(x) \ = w^T \emptyset(x) + b = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \quad (3.9)$$

SVM merupakan salah satu algoritma paling populer yang digunakan dalam klasifikasi. Algoritma ini memiliki beberapa keunggulan antara lain tidak memiliki kondisi minimum lokal, kemampuan generalisasi yang tinggi, dan mampu beradaptasi pada jumlah data sampel yang kecil dan data sampel dengan dimensi tinggi (Abdi, Hosseini, & Rezghi, 2012). (Liu, Yu, Xiangji, & An, 2011) menyebutkan bahwa SVM standar dapat bekerja dengan baik dalam dataset yang memiliki rasio ketidakseimbangan kecil dan sedang.

Kinerja SVM dapat ditingkatkan dengan mengintegrasikannya dengan metode pengurangan dimensi dan teknik optimalisasi parameter. Thaseen dan Kumar (Sumaiya Thaseen & Aswani Kumar, 2017) menunjukkan bahwa SVM dengan reduksi dimensi dan optimasi parameter kernel dapat meningkatkan kinerja klasifikasi dan mengurangi waktu klasifikasi. Penggunaan metode normalisasi juga sangat berpengaruh dalam proses pemilihan fitur dan implementasi SVM pada IDS. Normalisasi dapat mempersingkat tahapan pembelajaran dan meningkatkan kinerja pengklasifikasi (Li & Liu, 2011).

SVM awalnya hanya didisain untuk klasifikasi dua kelas, kemudian dikembangkan untuk klasifikasi muti-kelas. Ada dua metode yang umum digunakan untuk mengimplementasikan SVM pada multi-kelas, yaitu menggunakan metode *one-against-all* (OAA) atau metode *one-against-one* (OAO). Studi yang dilakukan oleh Hsu dan Lin (Hsu & Lin, 2002) menunjukkan bahwa metode OAO mempunyai keunggulan dalam penggunaan praktis. Pada metode ini,

model SVM multi-kelas dengan dataset k kelas akan dibangun dari $k(k-1)/2$ SVM. Masing-masing melakukan pelatihan menggunakan dataset dua kelas.

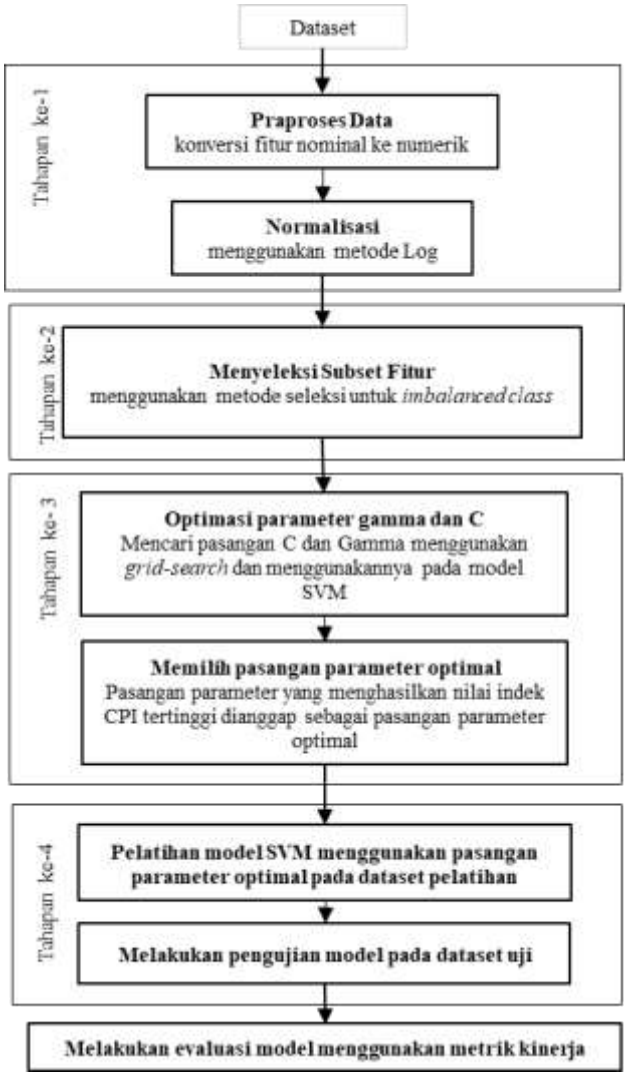


Gambar 3.5 Struktur SVM-OP multi-kelas menggunakan metode OAO

Pada penelitian ini yang membedakan lima jenis lalulintas jaringan, kami membangun sepuluh SVM seperti yang terlihat pada Gambar 3.4. Lima jenis lalulintas jaringan tersebut adalah *Normal*, *Probe*, *Denial of Service* (DoS), *User to Root* (U2R), dan *Remote to Local* (R2L). SVM-1 digunakan untuk kelas Normal dan DoS, SVM-2 untuk kelas Normal dan R2L, SVM-3 untuk kelas Normal dan Probe, dan SVM-4 untuk kelas Normal dan U2R. Sedangkan SVM-5 untuk kelas DoS dan R2L, SVM-6 untuk kelas DoS dan Probe, SVM-7 untuk kelas DoS dan U2R, SVM-8 untuk kelas R2L dan Probe, SVM-9 untuk kelas R2L dan Probe, dan SVM-10 untuk kelas Probe dan U2R.

Diagram proses pemodelan IDS berbasis SVM-OP yang dibuat pada penelitian ini dapat dilihat pada Gambar 3.6. Secara keseluruhan pemodelan dibagi menjadi empat tahapan, dimana tahapan pertama merupakan tahapan praproses data, pada tahapan kedua dilakukan proses seleksi fitur menggunakan metode seleksi untuk *imbalanced class* RIGFS dan RWIGFS, pada tahapan ketiga dilakukan proses optimasi parameter kernel RBF, dan pada tahapan keempat dilakukan proses pelatihan dan pengujian model SVM-OP dengan parameter optimal yang dihasilkan pada tahapan sebelumnya menggunakan dataset yang diuji. Uji coba pelatihan dan pengujian pada dataset Kyoto2006++ dilakukan dengan metoda 10-*fold cross validation.* Sedangkan pada dataset NSL-KDD, uji coba dilakukan dengan membagi 70% untuk data pengujian dan 30% untuk data

pelatihan. Pengujian model dilakukan dengan menggunakan satuan ukur *accuracy*, *sensitivity*, dan *specificity*.



Gambar 3.6 Diagram pemodelan IDS berbasis SVM-OP

Pencarian nilai optimal parameter kernel RBF pada penelitian ini dilakukan menggunakan metode *grid-search* dan metode pengujian 5-*fold cross validation* pada SVM. Kami menggunakan SVM dari modul Lib-SVM di aplikasi data mining WEKA versi 3.8.3 (Waikato University, 2018). Nilai parameter C yang digunakan sebanyak 11 buah yang dibuat dengan skala logarithmic antara 0.001 dan 100000, yaitu:

$10^{-3.0}, 10^{-2.2}, 10^{-1.4}, 10^{-0.6}, 10^{0.2}, 10^{1.0}, 10^{1.8}, 10.^{2.6}, 10^{3.4}, 10^{4.2}, 10^{5.0}$.

Nilai gamma yang digunakan sebanyak 11 buah yang dibuat dengan skala logaritmik antara 0.001 dan 1.5, yaitu:

$10^{-3.0}, 10^{-2.7}, 10^{-2.4}, 10^{-2.0}, 10^{-1.7}, 10^{-1.4}, 10^{-1.1}, 10.^{-0.8}, 10^{-0.5}, 10^{-0.1}, 10^{0.2}$.

Proses optimasi parameter kernel RBF dievaluasi menggunakan CPI. Parameter optimal direpresentasikan dengan nilai CPI tertinggi yang dicapai dalam pengujian. Hasil proses optimasi parameter kernel RBF pada dataset NSL-KDD dengan metode seleksi fitur RIGFS dan RWIGFS disajikan pada Tabel 3.5.

Tabel 3.5 Hasil optimasi parameter C dan gamma

Metode Seleksi Fitur	SVM.C	SVM.gamma
RWIGFS 18 fitur	398.107171 atau $10^{2.6}$	0.167210 atau $10^{-0.8}$
MRIGFS 19 fitur	2511.886432 atau $10^{3.4}$	0.038729 atau $10^{-1.4}$

Nilai optimal untuk pasangan parameter C dan gamma yang didapatkan dari proses optimasi menggunakan dataset NSL-KDD dan metode seleksi fitur MRIGFS adalah 398.107170 dan 0.038729. Sedangkan untuk dataset NSL-KDD dengan metode seleksi fitur RWIGFS adalah 63.095734 dan 0.167210.

3.7. Pemodelan IDS berbasis SVM-OW

Cost-sensitive learning merupakan strategi penanganan *imbalanced-class* yang melakukan penyeimbangan distribusi kelas pada level algoritma sehingga tidak merubah kondisi dari data sampel. *Weighted-SVM* (WSVM) merupakan SVM yang mengimplementasikan *cost-sensitive learning*. Yang et al. (Yang, Song, & Wang, 2007) menyatakan bahwa WSVM dapat memperbaiki *sensitivity* SVM standar terhadap *outlier* pada klasifikasi data dua kelas.

WSVM menggunakan pendekatan yang menganggap setiap titik data memiliki bobot yang berbeda sesuai dengan kepentingan relatif pada kelasnya. Pendekatan ini membuat berbagai titik data memiliki kontribusi yang berbeda selama proses pembelajaran untuk pengambilan keputusan. Algoritma ini membangun sebuah *cost function* untuk mengurangi kesalahan klasifikasi dan memaksimalkan margin pemisahan.

Berbeda dengan istilah penalti dalam SVM standar, WSVM memberlakukan bobot hukuman untuk meminimalkan efek dari titik data yang kurang penting seperti *noise* dan *outlier*. Algoritma ini memberikan bobot W_i pada titik data x_i. Di SVM standar, nilai dari C adalah tetap dan semua titik data pelatihan diperlakukan sama selama proses pelatihan.

Pendekatan *support vector* membutuhkan solusi terkait optimasi. Misal vektor pelatihan x_i dipetakan oleh fungsi ϕ ke dalam ruang yang dimensinya lebih tinggi. Variabel *slack* ξ_i digunakan untuk mengukur penyimpangan data pelatihan. Dimana parameter C merupakan parameter yang nilainya ditentukan pengguna, yang berfungsi untuk menyeimbangkan antara kompleksitas solusi (*solution complexity*) dan kesalahan solusi (*solution error*). Formula optimasi untuk WSVM ditunjukkan pada persamaan (3.10), sedangkan untuk SVM standar ditunjukkan pada persamaan (3.11). Sehingga bisa disimpulkan bahwa perbedaan antara SVM dan WSVM ada pada *solution error* untuk meminimalkan kesalahan pelatihan. Untuk pelatihan WSVM training, nilai *solution error* $= \sum_{i=1}^{n} W_i \xi_i$ sedangkan pada SVM standar nilai *solution error* $= \sum_{i=1}^{n} \xi_i$.

$$Minimize\ \Phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} W_i \xi_i$$

Dimana:

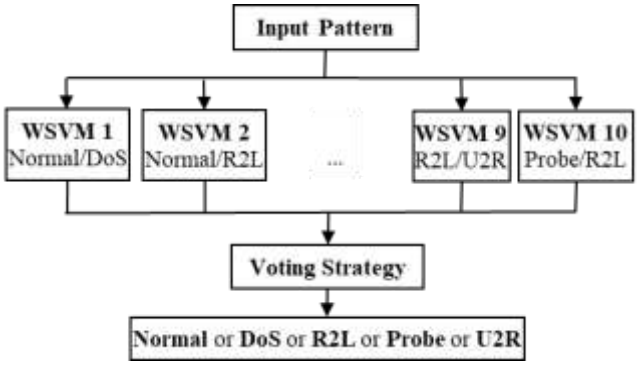
$$y_i \left(\left(w, \phi(x_i) \right) + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, N \qquad (3.10)$$
$$\xi_i \geq 0, \quad i = 1, \dots, N$$

$$Minimize\ \Phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$

Dimana:

$$y_i \left(\left(w, \phi(x_i) \right) + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, N \qquad (3.11)$$
$$\xi_i \geq 0, \quad i = 1, \dots, N$$

Pada aplikasi multi-kelas, model IDS ini juga menggunakan metode *one-against-one* (OAO) yang strukturnya dapat dilihat pada Gambar 3.7.



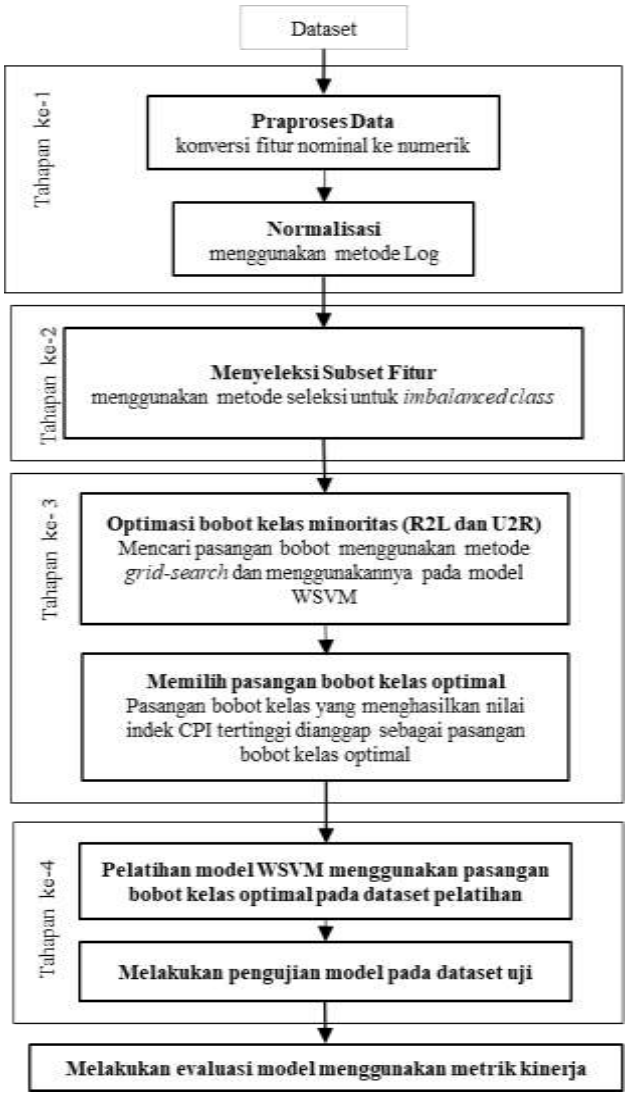
Gambar 3.7 Struktur WSVM multi-kelas menggunakan metode OAO

Diagram proses pemodelan IDS berbasis SVM-OW ditunjukkan pada Gambar 3.8. Secara keseluruhan pemodelan dibagi menjadi empat tahapan, Tahapan pertama merupakan tahapan praproses data, pada tahapan kedua dilakukan proses seleksi fitur menggunakan metode seleksi untuk *imbalanced class* RIGFS dan RWIGFS, tahapan ketiga merupakan tahapan optimasi bobot kelas dari dataset yang diuji, dan tahapan keempat merupakan tahapan pelatihan dan pengujian model SVM-OW pada dataset yang diuji dengan menggunakan parameter bobot kelas optimal yang dihasilkan pada tahapan sebelumnya.

Uji coba pelatihan dan pengujian pada dataset Kyoto2006++ dilakukan dengan metoda 10-*fold cross validation.* Sedangkan pada dataset NSL-KDD, uji coba dilakukan dengan membagi 70% untuk data pengujian dan 30% untuk data pelatihan. Selanjutnya dilakukan pengujian model menggunakan satuan ukur *accuracy*, *sensitivity*, dan *specificity.*

Pada dataset NSL-KDD, optimasi bobot kelas dilakukan dengan dua cara. Pertama, optimasi bobot dilakukan dengan menggunakan perbandingan bobot berdasarkan jumlah data pada masing-masing kelas serangan (OB-1). Kedua, optimasi bobot dilakukan pada kelas R2L dan U2R yang merupakan *minority class* dengan menggunakan *grid-search* (OB-2). Pencarian bobot optimal dilakukan pada rentang nilai 1 sampai dengan 10. Sedangkan bobot kelas lainnya ditetapkan dengan nilai 1. Hal ini dilakukan dengan tujuan untuk meningkatkan kinerja deteksi pada *minority class* dan mempertahankan kinerja kelas lainnya tetap tinggi. Proses optimasi bobot kelas dievaluasi menggunakan indek CPI. Pasangan bobot kelas

optimal direpresentasikan dengan nilai indek CPI tertinggi yang dicapai dalam pengujian. Untuk uji coba pada dataset Kyoto2006++ kami hanya menggunakan optimasi bobot pertama (OB-1).



Gambar 3.8 Diagram pemodelan IDS berbasis SVM-OW

Hasil optimasi bobot kelas SVM-OW pada dataset NSL-KDD dengan metode seleksi fitur MRIGFS dan RWIGFS disajikan pada Tabel 3.6. Nilai komposisi bobot kelas (Normal:DoS:R2L:Probe:U2R) melalui pendekatan perbandingan bobot berdasarkan jumlah data pada masing-masing kelas serangan (OB-1) adalah (2:2:101:9:1923). Melalui pendekatan OB-2 atau menggunakan

grid-search, nilai optimal komposisi bobot kelas yang didapat untuk subset data hasil seleksi fitur menggunakan MRIGFS adalah (1:1:5:1:10). Sedangkan untuk subset data hasil seleksi fitur menggunakan RWIGFS adalah (1:1:3:1:10).

Tabel 3.6 Hasil optimasi bobot kelas

Metode Seleksi Fitur	OB-1	OB-2
MRIGFS 19 fitur	2:2:101:9:1923	1:1:5:1:10
RWIGFS 18 fitur	2:2:101:9:1923	1:1:3:1:10

Catatan: Urutan bobot class [Normal : DoS : R2L : Probe : U2R]

3.8. Pemodelan IDS berbasis Ensemble-Voting

Penggabungan L-SCANN dengan SVM-OP dan SVM-OW menggunakan teknik *ensemble-voting* dilakukan untuk membangun model IDS berbasis CBC yang mampu melakukan *validasi* terhadap serangan yang diprediksi sebagai lalu lintas jaringan normal atau kondisi *False Negative.* Pada model ini, SVM-OP dan SVM-OW berperan sebagai validator untuk meminimalkan *False Negative* dan *False Positive.* Bagan pemodelan IDS berbasis *ensemble-voting* ditunjukkan pada Gambar 3.1.

Proses pelatihan dan pengujian dilakukan dengan metoda 10-*fold cross validation*. Pada setiap tahapan *fold*, output prediksi dari L-SCANN, SVM-OP, dan SVM-OW akan diproses *voting* untuk menghasilkan prediksi akhir. Output prediksi dari L-SCANN didapatkan dari file log yang dihasilkan dari program L-SCANN. Sedangkan output prediksi dari SVM-OP dan SVM-OW didapatkan dari file log yang dihasilkan dari modul Lib-SVM pada aplikasi data mining WEKA 3.8.3. Ketiga output prediksi tersebut selanjutnya ditabelkan secara manual dalam format file csv dan diproses menggunakan aplikasi Excel dari Microsoft Office.

3.9. Evaluasi Kinerja

Uji coba untuk mengevaluasi kinerja model IDS dilakukan dengan menggunakan dataset NSL-KDD 100% dan dataset Kyoto 2006++. Satuan ukur yang digunakan untuk mengukur kinerja model IDS pada penelitian ini adalah

accuracy, *sensitivity*, *specificity*, *false alarm rate* (FAR), dan *false negative rate* (FNR) yang berbasis *confusion matrix*. Dimana elemen dasar dari *confusion matrix* adalah kondisi *true positive* (TP), *true negative* (TN), *false positive* (FP) dan *false negative* (FN). Dimana TP menunjukkan jumlah sampel serangan yang diprediksi dengan benar sebagai serangan. TN mengacu pada jumlah sampel aktivitas jaringan normal yang diklasifikasikan dengan benar sebagai aktivitas jaringan normal. FP didefinisikan sebagai jumlah sampel aktivitas jaringan normal yang secara keliru diklasifikasikan sebagai serangan, FN mengacu pada jumlah sampel serangan yang secara keliru ditetapkan sebagai aktivitas jaringan normal.

Pada penelitian ini, *accuracy* didefinisikan sebagai tingkat kemampuan dari metode deteksi dalam mengenali aktivitas serangan dan aktivitas jaringan normal dengan benar, perhitungannya dilakukan dengan persamaan (3.12). *Sensitivity* didefinisikan sebagai tingkat kemampuan dari metode deteksi dalam mengenali setiap aktivitas serangan, perhitungannya dilakukan dengan persamaan (3.13). Sedangkan *specificity* didefinisikan sebagai tingkat kemampuan dari metode deteksi dalam mengenali setiap aktivitas jaringan normal, perhitungannya dilakukan dengan persamaan (3.14).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.12}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.13}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.14}$$

$$FAR = \frac{FP}{FP + TN} = 1 - Specificity \tag{3.15}$$

$$FNR = \frac{FN}{FN + TP} = 1 - Sensitivity \tag{3.16}$$

FAR didefinisikan sebagai ratio terjadinya kesalahan deteksi aktivitas jaringan normal yang diprediksi sebagai serangan, perhitungannya dilakukan

dengan persamaan (3.15). FNR yang didefinisikan sebagai ratio terjadinya kesalahan deteksi dimana serangan dianggap sebagai aktivitas jaringan normal, perhitungannya dilakukan dengan persamaan (3.16).

Halaman ini sengaja dikosongkan

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini disajikan hasil uji coba dan pembahasan untuk masalah normalisasi, seleksi fitur, pemodelan IDS berbasis CBC, pemodelan IDS berbasis SVM-OP, pemodelan IDS berbasis SVM-OW, dan penggabungan ketiga model IDS tersebut menggunakan metode *ensemble-voting*.

4.1. Normalisasi

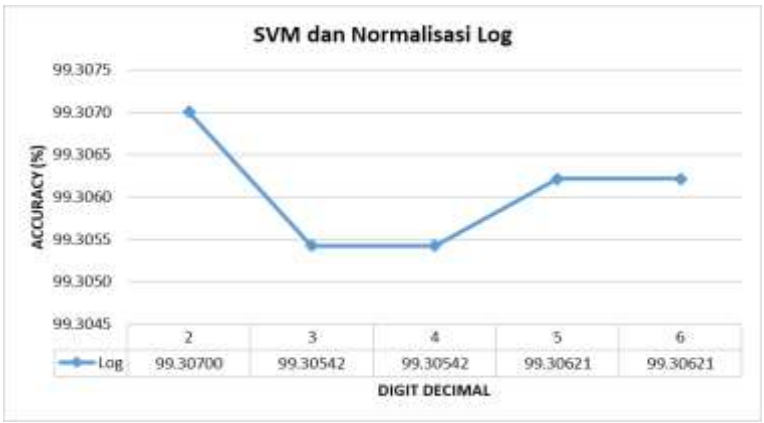
Normalisasi atribut adalah langkah penting yang harus dilakukan ketika kita berhadapan dengan dataset yang berisi variabel dengan unit dan rentang nilai yang berbeda. Normalisasi akan mengubah semua nilai fitur untuk berada dalam skala atau kisaran kecil yang ditentukan. Pada penelitian ini kami melakukan tiga analisa terkait dengan normalisasi, yaitu: a) analisa pengaruh pembulatan hasil normalisasi terhadap kinerja pengklasifikasi, b) analisa pengaruh penggunaan metode normalisasi terhadap hasil seleksi fitur, dan c) analisa perubahan nilai *mutual information´fitur* akibat pembulatan hasil normalisasi. Kami menggunakan tiga metode normalisasi yaitu Min-max, Z-score, dan Log.

a) Analisis pengaruh pembulatan hasil normalisasi terhadap kinerja pengklasifikasi

Untuk mengetahui pengaruh penggunaan metode normalisasi dan jumlah digit yang digunakan dalam proses pembulatan hasil normalisasi terhadap kinerja pengklasifikasi, kami melakukan uji coba menggunakan dataset pelatihan NSL-KDD 100% dan dua pengklasifikasi yaitu SVM dan k-NN. Kami memproses fitur-fitur dalam NSL-KDD dengan tiga metode normalisasi: Min-max, Z-score, dan Log. Hasil normalisasi dari ketiga metode tersebut kami proses pembulatan dengan digit desimal dari 2 sampai dengan 11. Hasil uji coba kami sajikan pada Gambar 4.1, 4.2, 4.3, dan 4.4. Gambar 4.1 menunjukkan bahwa jumlah digit desimal yang digunakan dalam pembulatan hasil normalisasi pada metode Min-max, Z-score, dan Log berpengaruh terhadap *accuracy* dari pengklasifikasi SVM.

(a)

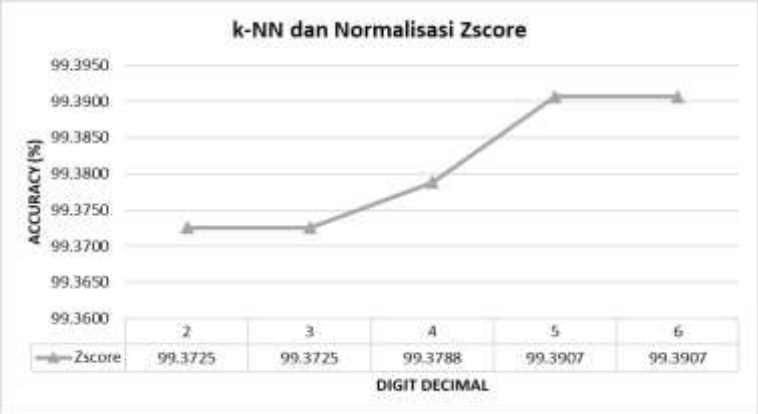
(b)



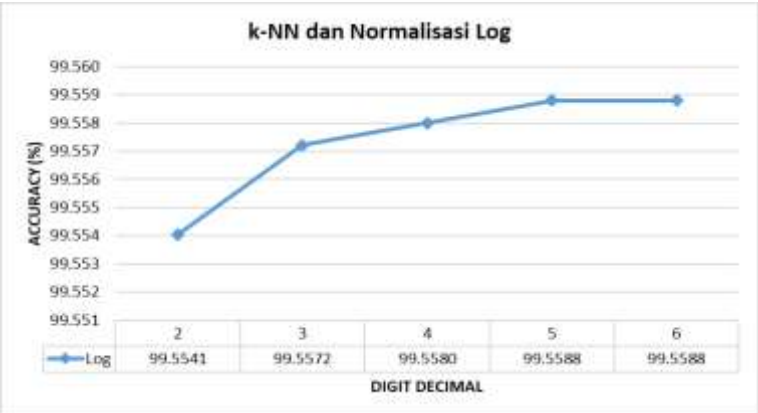
(c)

Gambar 4.1 Grafik *accuracy* pengklasifikasi SVM terhadap jumlah digit desimal yang digunakan dalam pembulatan normalisasi: (a) Min-max, (b) Z-score, dan (c) Log pada dataset NSL-KDD

(a)



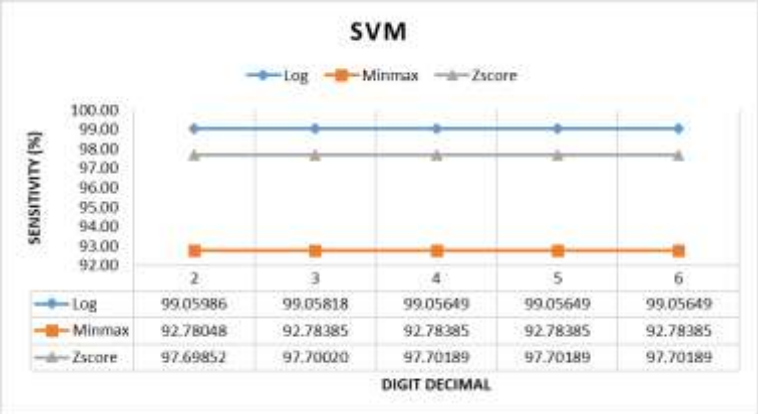
(b)



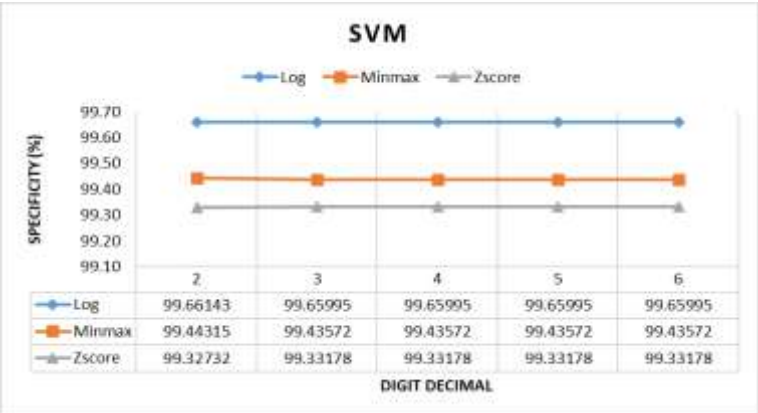
(c)

Gambar 4.2 Grafik *accuracy* pengklasifikasi k-NN terhadap jumlah digit desimal yang digunakan dalam pembulatan normalisasi: (a) Min-max, (b) Z-score, dan (c) Log pada dataset NSL-KDD

(a)



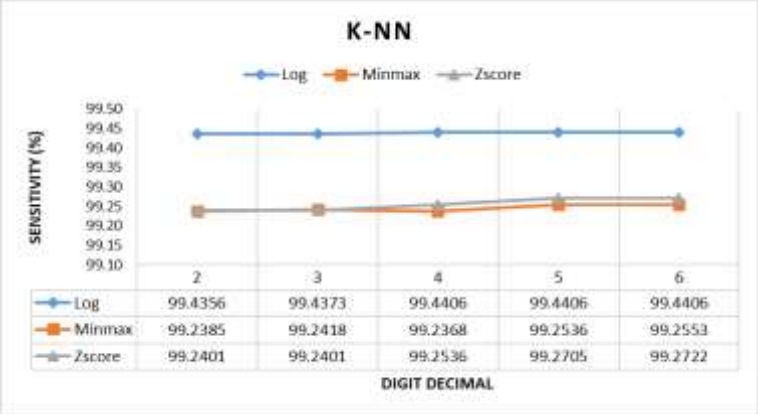
(b)



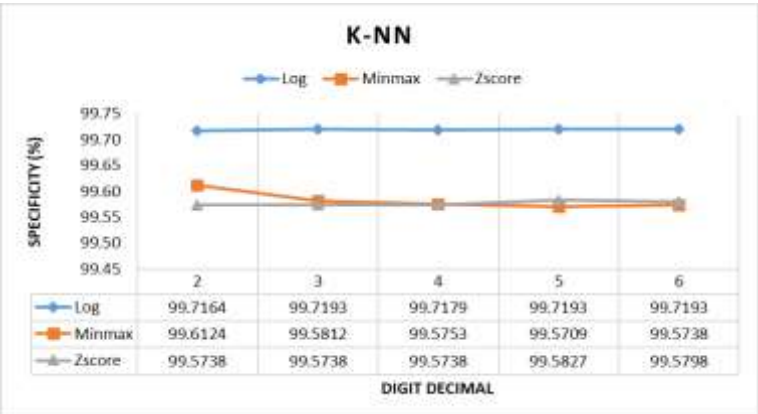
(c)

Gambar 4.3 Grafik perbandingan pengaruh penggunaan metode normalisasi Min-max, Z-score, dan Log pada dataset NSL-KDD pada (a) *accuracy*, (b) *sensitivity*, dan (c) *specificity* pengklasifikasi SVM

(a)



(b)



(c)

Gambar 4.4 Grafik perbandingan pengaruh penggunaan metode normalisasi Min-max, Z-score, dan Log pada dataset NSL-KDD pada (a) *accuracy*, (b) *sensitivity*, dan (c) *specificity* pengklasifikasi k-NN

Sedangkan Gambar 4.2 menunjukkan pengaruh ketiga metode normalisasi terhadap *accuracy* dari pengklasifikasi k-NN. Gambar 4.3 dan 4.4 menunjukkan bahwa metode Log menghasilkan *accuracy*, *sensitivity*, dan *specificity* yang lebih tinggi dibanding metode normalisasi Min-max dan Z-score pada pengklasifikasi SVM dan k-NN. Hasil ini sama dengan yang diperoleh pada (Said et al., 2011).

Gambar 4.5, 4.6, dan 4.7 menunjukkan bahwa penggunaan metode normalisasi juga berpengaruh terhadap kecepatan proses pengklasifikasi. Gambar 4.5 menunjukkan bahwa penggunaan SVM dengan metode normalisasi Log menghasilkan kecepatan yang lebih baik dibanding menggunakan metode normalisasi Min-max dan Z-score. Gambar 4.6 dan 4.7 menunjukkan kondisi yang sama ketika diimplementasikan pada pengklasifikasi k-NN dan L-SCANN.

Gambar 4.5 Grafik perbandingan pengaruh penggunaan metode normalisasi pada dataset NSL-KDD terhadap kecepatan proses SVM

Gambar 4.6 Grafik perbandingan pengaruh penggunaan metode normalisasi pada dataset NSL-KDD terhadap kecepatan proses k-NN

Gambar 4.7 Grafik perbandingan pengaruh penggunaan metode normalisasi pada dataset NSL-KDD terhadap kecepatan proses L-SCANN

Dari hasil uji coba tersebut kami dapat mengambil dua kesimpulan. Pertama, kinerja dari pengklasifikasi SVM dan k-NN dipengaruhi oleh jenis metode normalisasi dan jumlah digit yang digunakan pada proses pembulatan hasil normalisasi. Kedua, penggunaan metode normalisasi Log pada SVM dan k-NN menghasilkan *accuracy*, *sensitivity*, *specificity*, dan kecepatan proses klasifikasi yang lebih baik dibanding metode normalisasi Min-max dan Z-score.

b) Analisis pengaruh penggunaan metode normalisasi terhadap hasil seleksi fitur

Selanjutnya untuk mengetahui pengaruh penggunaan metode normalisasi dan pembulatan hasil normalisasi, kami melakukan uji coba dengan menggunakan metode seleksi fitur yang menggunakan algoritma Ranker untuk pencarian dan Information-Gain sebagai *attribute evaluation*. Metode ini dikenal dengan metode seleksi fitur Rank+InformationGain. Uji coba dilakukan pada dataset NSL-KDD 100%. Untuk menyederhanakan pengamatan, fitur-fitur yang digunakan hanya fitur-fitur numerik (*continuous*) pada dataset NSL-KDD. Karena metode normalisasi digunakan untuk memproses fitur-fitur numerik. Tabel 4.1 menampilkan daftar fitur-fitur numerik dari dataset NSL-KDD. Tabel 4.2, 4.3, dan 4.4 menampilkan hasil urutan fitur setelah proses seleksi fitur menggunakan metode Rank+InformationGain.

Tabel 4.1 Daftar fitur *continuous* pada dataset NSL-KDD

No Fitur	Nama Fitur
1	duration
2	src_bytes
3	dst_bytes
4	wrong_fragment
5	urgent
6	hot
7	num_failed_logins
8	num_compromised
9	root_shell
10	su_attempted
11	num_root
12	num_file_creations
13	num_shells
14	num_access_files
15	num_outbound_cmds
16	count
17	srv_count
18	serror_rate
19	srv_serror_rate
20	rerror_rate
21	srv_rerror_rate
22	same_srv_rate
23	diff_srv_rate
24	srv_diff_host_rate
25	dst_host_count
26	dst_host_srv_count
27	dst_host_same_srv_rate
28	dst_host_diff_srv_rate
29	dst_host_same_src_port_rate
30	dst_host_srv_diff_host_rate
31	dst_host_serror_rate
32	dst_host_srv_serror_rate
33	dst_host_rerror_rate
34	dst_host_srv_rerror_rate

Tabel 4.2 menunjukkan bahwa pada penggunaan jumlah digit desimal dari 6 sampai 11 untuk pembulatan hasil normalisasi Min-max menghasilkan urutan hasil fitur seleksi yang sama. Sedangkan untuk penggunaan di bawah 6 digit, menghasilkan urutan fitur hasil seleksi yang berbeda. Fitur-fitur pada 12 urutan teratas berbeda pada penggunaan jumlah digit desimal dari 2 sampai 6. Fitur-fitur dari urutan ke-13 sampai ke-22 berbeda pada penggunaan jumlah digit desimal dari 2 sampai 5. Fitur-fitur dari urutan ke-23 sampai ke-25 berbeda pada penggunaan jumlah digit desimal dari 2 sampai 4. Sedangkan untuk fitur-fitur dari urutan ke-26 sampai ke-30 mengalami perbedaan pada penggunaan 2 digit desimal, dan fitur-fitur pada urutan di atas 30 tidak mengalami perbedaan.

Tabel 4.2 Perbandingan urutan fitur hasil seleksi fitur dengan normalisasi Min-max

Rank	Jumlah digit untuk pembulatan hasil normalisasi Min-max									
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	***23***	***23***	***23***	***23***	***23***	2	2	2	2	2
2	***22***	***22***	***22***	***22***	***3***	23	23	23	23	23
3	***28***	***28***	***28***	***28***	***22***	3	3	3	3	3
4	***16***	***16***	***16***	***3***	***28***	22	22	22	22	22
5	***26***	***26***	***26***	***16***	***16***	28	28	28	28	28
6	***27***	***27***	***27***	***26***	***26***	16	16	16	16	16
7	***31***	***31***	***31***	***27***	***27***	26	26	26	26	26
8	***18***	***18***	***18***	***31***	***31***	27	27	27	27	27
9	***32***	***32***	***32***	***18***	***18***	31	31	31	31	31
10	***19***	***19***	***19***	***32***	***32***	18	18	18	18	18
11	***30***	***30***	***3***	***19***	***19***	32	32	32	32	32
12	***29***	***29***	***30***	***30***	***2***	19	19	19	19	19
13	***25***	***25***	***29***	***29***	30	30	30	30	30	30
14	***24***	***17***	***25***	***25***	29	29	29	29	29	29
15	***17***	***24***	***17***	***17***	25	25	25	25	25	25
16	***33***	***33***	***24***	***24***	17	17	17	17	17	17
17	***34***	***34***	***33***	***33***	24	24	24	24	24	24
18	***20***	***20***	***34***	***34***	33	33	33	33	33	33
19	***21***	***3***	***20***	***20***	34	34	34	34	34	34
20	***1***	***21***	***21***	***1***	20	20	20	20	20	20
21	***6***	1	1	***21***	1	1	1	1	1	1
22	4	6	6	***2***	21	21	21	21	21	21
23	3	4	2	6	6	6	6	6	6	6
24	***2***	***8***	4	4	4	4	4	4	4	4
25	***9***	***2***	8	8	8	8	8	8	8	8
26	***12***	11	11	11	11	11	11	11	11	11
27	***14***	9	9	9	9	9	9	9	9	9
28	***11***	12	12	12	12	12	12	12	12	12
29	***7***	14	14	14	14	14	14	14	14	14
30	***8***	7	7	7	7	7	7	7	7	7
31	10	10	10	10	10	10	10	10	10	10
32	5	5	5	5	5	5	5	5	5	5
33	13	13	13	13	13	13	13	13	13	13
34	15	15	15	15	15	15	15	15	15	15

Tabel 4.3 yang menampilkan hasil uji coba dengan metode normalisasi Z-score juga menunjukkan adanya kesamaan urutan fitur hasil seleksi jika menggunakan jumlah digit decimal dari 5 sampai 11 untuk pembulatan hasil normalisasi. Dan untuk penggunaan di bawah 5 digit menghasilkan urutan fitur hasil seleksi yang berbeda. Fitur-fitur pada 11 urutan teratas berbeda pada penggunaan jumlah digit desimal dari 2 sampai 4. Fitur-fitur dari urutan ke-12 sampai ke-22 berbeda pada penggunaan jumlah digit desimal dari 2 sampai 3. Sedangkan untuk fitur pada urutan ke-23 mengalami perbedaan pada penggunaan 2 digit desimal, dan fitur-fitur pada urutan di atas 23 tidak mengalami perbedaan. Terlihat perbedaan urutan yang dihasilkan pada metode normalisasi Z-score lebih sedikit dibandingkan jika menggunakan metode normalisasi Min-max.

Tabel 4.3 Perbandingan urutan fitur hasil seleksi fitur dengan normalisasi Z-score

Rank	Jumlah digit untuk pembulatan hasil normalisasi Zscore									
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	***23***	***23***	***23***	2	2	2	2	2	2	2
2	***22***	***22***	***22***	23	23	23	23	23	23	23
3	***28***	***28***	***28***	3	3	3	3	3	3	3
4	***16***	***16***	***3***	22	22	22	22	22	22	22
5	***26***	***26***	***16***	28	28	28	28	28	28	28
6	***27***	***3***	***26***	16	16	16	16	16	16	16
7	***31***	***27***	***27***	26	26	26	26	26	26	26
8	***18***	***31***	***31***	27	27	27	27	27	27	27
9	***32***	***18***	***18***	31	31	31	31	31	31	31
10	***19***	***32***	***32***	18	18	18	18	18	18	18
11	***30***	***19***	***2***	32	32	32	32	32	32	32
12	***29***	***30***	19	19	19	19	19	19	19	19
13	***25***	***29***	30	30	30	30	30	30	30	30
14	***3***	***25***	29	29	29	29	29	29	29	29
15	***17***	***17***	25	25	25	25	25	25	25	25
16	***24***	***24***	17	17	17	17	17	17	17	17
17	***33***	***33***	24	24	24	24	24	24	24	24
18	***34***	***34***	33	33	33	33	33	33	33	33
19	***20***	***20***	34	34	34	34	34	34	34	34
20	***21***	***1***	20	20	20	20	20	20	20	20
21	1	***21***	1	1	1	1	1	1	1	1
22	***6***	***2***	21	21	21	21	21	21	21	21
23	***2***	6	6	6	6	6	6	6	6	6
24	4	4	4	4	4	4	4	4	4	4
25	8	8	8	8	8	8	8	8	8	8
26	11	11	11	11	11	11	11	11	11	11
27	9	9	9	9	9	9	9	9	9	9
28	12	12	12	12	12	12	12	12	12	12
29	14	14	14	14	14	14	14	14	14	14
30	7	7	7	7	7	7	7	7	7	7
31	10	10	10	10	10	10	10	10	10	10
32	5	5	5	5	5	5	5	5	5	5
33	13	13	13	13	13	13	13	13	13	13
34	15	15	15	15	15	15	15	15	15	15

Kondisi yang berbeda terlihat pada Tabel 4.4 yang menunjukkan hasil urutan fitur dengan menggunakan metode normalisasi Log. Penggunaan jumlah digit desimal dari 2 sampai 11 untuk pembulatan hasil normalisasi dengan metode Log tidak mempengaruhi urutan hasil fitur seleksi.

Dari hasil uji coba tersebut kami mengambil kesimpulan bahwa penggunaan digit desimal yang berbeda untuk pembulatan hasil normalisasi dapat mempengaruhi urutan fitur hasil seleksi jika menggunakan metode normalisasi Min-max dan Z-score. Sedangkan pada metode normalisasi Log, tidak menyebabkan perubahan urutan fitur.

Tabel 4.4 Perbandingan urutan fitur hasil seleksi fitur dengan normalisasi Log

Rank	Jumlah digit untuk pembulatan hasil normalisasi Log									
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	2	2	2	2	2	2	2	2	2	2
2	23	23	23	23	23	23	23	23	23	23
3	3	3	3	3	3	3	3	3	3	3
4	22	22	22	22	22	22	22	22	22	22
5	28	28	28	28	28	28	28	28	28	28
6	16	16	16	16	16	16	16	16	16	16
7	26	26	26	26	26	26	26	26	26	26
8	27	27	27	27	27	27	27	27	27	27
9	31	31	31	31	31	31	31	31	31	31
10	18	18	18	18	18	18	18	18	18	18
11	32	32	32	32	32	32	32	32	32	32
12	19	19	19	19	19	19	19	19	19	19
13	30	30	30	30	30	30	30	30	30	30
14	29	29	29	29	29	29	29	29	29	29
15	25	25	25	25	25	25	25	25	25	25
16	17	17	17	17	17	17	17	17	17	17
17	24	24	24	24	24	24	24	24	24	24
18	33	33	33	33	33	33	33	33	33	33
19	34	34	34	34	34	34	34	34	34	34
20	20	20	20	20	20	20	20	20	20	20
21	1	1	1	1	1	1	1	1	1	1
22	21	21	21	21	21	21	21	21	21	21
23	6	6	6	6	6	6	6	6	6	6
24	4	4	4	4	4	4	4	4	4	4
25	8	8	8	8	8	8	8	8	8	8
26	11	11	11	11	11	11	11	11	11	11
27	9	9	9	9	9	9	9	9	9	9
28	12	12	12	12	12	12	12	12	12	12
29	14	14	14	14	14	14	14	14	14	14
30	7	7	7	7	7	7	7	7	7	7
31	10	10	10	10	10	10	10	10	10	10
32	5	5	5	5	5	5	5	5	5	5
33	13	13	13	13	13	13	13	13	13	13
34	15	15	15	15	15	15	15	15	15	15

c) Analisis perubahan nilai mutual information akibat pembulatan hasil normalisasi

Selanjutnya kami melakukan analisis lebih detail terhadap nilai *information-gain* dari fitur-fitur ketika dilakukan pembulatan hasil normalisasi dengan digit desimal yang berbeda. Dimana nilai *information-gain* merepresentasikan nilai *mutual information* dari fitur dengan label klasifikasi. Dengan hipotesa awal bahwa pembulatan hasil normalisasi dengan digit yang berbeda dapat mempengaruhi nilai *information-gain* dari fitur yang diproses. Fitur-fitur yang dianalisis hanya fitur-fitur numerik (*continuous*) pada dataset NSL-KDD.

Untuk memudahkan melakukan pengamatan apakah ada perubahan nilai *mutual information* akibat pembulatan hasil normalisasi, kami menghitung dan menyajikan nilai *information-gain* dalam bentuk tabel. Tabel 4.5, 4.6, dan 4.7 secara berturutan menampilkan nilai *information-gain* dari proses normalisasi dengan metode Min-max, Z-score dan Log. Kami membandingkan nilai *information-gain* dari fitur-fitur sebelum normalisasi dengan nilainya setelah normalisasi. Nilai *information-gain* yang berbeda kami beri warna biru.

Tabel 4.5 Nilai Information-Gain fitur hasil normalisasi Min-max

	Fitur		**Nilai Information-Gain Fitur Hasil Normalisasi Min-max**								
	No	**Nama**	**Original**	**2 digit**	**3 digit**	**4 digit**	**5 digit**	**6 digit**	**7 digit**	**8 digit**	**9 digit**
1	1	***duration***	0.0783	***0.0386***	***0.0493***	***0.0617***	0.0783	0.0783	0.0783	0.0783	0.0783
2	5	***src_bytes***	1.0311	***0.0037***	***0.0066***	***0.0151***	***0.0459***	***0.4057***	***0.7749***	***0.9465***	1.0311
3	6	***dst_bytes***	0.6618	***0.0100***	***0.0944***	***0.4192***	***0.6431***	0.6577	0.6618	0.6618	0.6618
4	8	wrong_fragment	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130
5	9	urgent	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
6	10	hot	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334
7	11	num_failed_logins	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
8	13	***num_compromised***	0.0109	***0.0017***	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109
9	14	root_shell	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031
10	15	su_attempted	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
11	16	***num_root***	0.0047	***0.0029***	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047
12	17	num_file_creations	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030
13	18	num_shells	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
14	19	num_access_files	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029
15	20	num_outbound_cmds	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16	23	***count***	0.6046	***0.5862***	0.6046	0.6046	0.6046	0.6046	0.6046	0.6046	0.6046
17	24	srv_count	0.2351	0.1972	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351
18	25	serror_rate	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483
19	26	srv_serror_rate	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145
20	27	rerror_rate	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091
21	28	srv_rerror_rate	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764
22	29	same_srv_rate	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571
23	30	diff_srv_rate	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255
24	31	srv_diff_host_rate	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136
25	32	***dst_host_count***	0.3004	***0.2956***	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004
26	33	***dst_host_srv_count***	0.5928	***0.5826***	0.5928	0.5928	0.5928	0.5928	0.5928	0.5928	0.5928
27	34	dst_host_same_srv_rate	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764
28	35	dst_host_diff_srv_rate	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531
29	36	dst_host_same_src_port_rate	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434
30	37	dst_host_srv_diff_host_rate	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774
31	38	dst_host_serror_rate	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659
32	39	dst_host_srv_serror_rate	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381
33	40	dst_host_rerror_rate	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353
34	41	dst_host_srv_rerror_rate	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200
Total			**9.7334**	**7.9320**	**8.1125**	**8.4582**	**8.7295**	**9.1039**	**9.4772**	**9.6487**	**9.7334**
Selisih dengan Total Information-Gain fitur sebelum normalisasi				**1.8014**	**1.6209**	**1.2752**	**1.0039**	**0.6295**	**0.2562**	**0.0846**	**-**

Pada Tabel 4.5 yang menyajikan hasil normalisasi dengan metode Min-max tampak ada delapan fitur yang mengalami perubahan *information-gain* untuk pembulatan hasil normalisasi dengan tempat desimal di bawah 9 digit. Delapan fitur itu adalah 1) duration, 2) src_bytes, 3) dst_bytes, 4) num_compromised, 5) num_root, 6) count, 7) dst_host_count, dan 8) dst_host_srv_count.

Tabel 4.6 Nilai Information-Gain fitur hasil normalisasi Z-score

	Fitur		**Nilai Informatiion-Gain Fitur Hasil Normalisasi Z-score**								
	No	**Nama**	**Original**	**2 digit**	**3 digit**	**4 digit**	**5 digit**	**6 digit**	**7 digit**	**8 digit**	**9 digit**
1	1	***duration***	0.0783	***0.0491***	0.0783	0.0783	0.0783	0.0783	0.0783	0.0783	0.0783
2	5	***src_bytes***	1.0311	***0.0193***	***0.0483***	***0.5261***	***0.7957***	***0.9540***	1.0311	1.0311	1.0311
3	6	***dst_bytes***	0.6618	***0.2973***	***0.5781***	***0.6496***	0.6618	0.6618	0.6618	0.6618	0.6618
4	8	wrong_fragment	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130
5	9	Urgent	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
6	10	Hot	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334
7	11	num_failed_logins	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
8	13	num_compromised	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109
9	14	root_shell	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031
10	15	su_attempted	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
11	16	num_root	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047
12	17	num_file_creations	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030
13	18	num_shells	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
14	19	num_access_files	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029
15	20	num_outbound_cmds	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16	23	***count***	0.6046	***0.6045***	0.6046	0.6046	0.6046	0.6046	0.6046	0.6046	0.6046
17	24	srv_count	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351
18	25	serror_rate	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483
19	26	srv_serror_rate	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145
20	27	rerror_rate	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091
21	28	srv_rerror_rate	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764
22	29	same_srv_rate	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571
23	30	diff_srv_rate	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255
24	31	srv_diff_host_rate	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136
25	32	dst_host_count	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004
26	33	***dst_host_srv_count***	0.5928	***0.5921***	0.5928	0.5928	0.5928	0.5928	0.5928	0.5928	0.5928
27	34	dst_host_same_srv_rate	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764
28	35	dst_host_diff_srv_rate	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531
29	36	dst_host_same_src_port_rate	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434
30	37	dst_host_srv_diff_host_rate	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774
31	38	dst_host_serror_rate	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659
32	39	dst_host_srv_serror_rate	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381
33	40	dst_host_rerror_rate	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353
34	41	dst_host_srv_rerror_rate	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200
Total			**9.7334**	**8.3271**	**8.6669**	**9.2161**	**9.4980**	**9.6562**	**9.7334**	**9.7334**	**9.7334**
Selisih dengan Total Information-Gain fitur sebelum normalisasi				**1.4063**	**1.0665**	**0.5172**	**0.2354**	**0.0771**	-	-	-

Pada Tabel 4.6 yang menyajikan hasil normalisasi dengan metode Z-score tampak ada lima fitur yang mengalami perubahan *information-gain* untuk pembulatan hasil normalisasi dengan pembulatan di bawah 4 digit desimal. Lima fitur itu adalah 1) duration, 2) src_bytes, 3) dst_bytes, 4) count, dan 5) dst_host_srv_count.

Tabel 4.7 Nilai Information-Gain fitur hasil normalisasi Log

	Fitur		**Nilai Information-Gain Fitur Hasil Normalisasi Log**								
	No	**Nama**	**Original**	**2 digit**	**3 digit**	**4 digit**	**5 digit**	**6 digit**	**7 digit**	**8 digit**	**9 digit**
1	1	***duration***	0.0783	***0.0786***	0.0783	0.0783	0.0783	0.0783	0.0783	0.0783	0.0783
2	5	***src_bytes***	1.0311	***1.0183***	***1.0305***	1.0311	1.0311	1.0311	1.0311	1.0311	1.0311
3	6	***dst_bytes***	0.6618	***0.6549***	***0.6609***	0.6618	0.6618	0.6618	0.6618	0.6618	0.6618
4	8	wrong_fragment	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130
5	9	urgent	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
6	10	hot	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334	0.0334
7	11	num_failed_logins	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
8	13	num_compromised	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109
9	14	root_shell	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031
10	15	su_attempted	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
11	16	num_root	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047	0.0047
12	17	num_file_creations	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030
13	18	num_shells	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
14	19	num_access_files	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029
15	20	num_outbound_cmds	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16	23	***count***	0.6046	***0.6040***	0.6046	0.6046	0.6046	0.6046	0.6046	0.6046	0.6046
17	24	srv_count	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351	0.2351
18	25	***serror_rate***	0.5483	***0.5476***	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483	0.5483
19	26	srv_serror_rate	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145	0.5145
20	27	rerror_rate	0.1091	0.1089	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091	0.1091
21	28	srv_rerror_rate	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764	0.0764
22	29	***same_srv_rate***	0.6571	***0.6543***	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571	0.6571
23	30	***diff_srv_rate***	0.7255	***0.7247***	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255	0.7255
24	31	srv_diff_host_rate	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136
25	32	dst_host_count	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004	0.3004
26	33	dst_host_srv_count	0.5928	0.5928	0.5928	0.5928	0.5928	0.5928	0.5928	0.5928	0.5928
27	34	***dst_host_same_srv_rate***	0.5764	***0.5765***	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764	0.5764
28	35	dst_host_diff_srv_rate	0.6531	0.6532	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531
29	36	***dst_host_same_src_port_rate***	0.3434	***0.3394***	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434	0.3434
30	37	***dst_host_srv_diff_host_rate***	0.3774	***0.3755***	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774	0.3774
31	38	***dst_host_serror_rate***	0.5659	***0.5649***	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659	0.5659
32	39	***dst_host_srv_serror_rate***	0.5381	***0.5380***	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381	0.5381
33	40	***dst_host_rerror_rate***	0.1353	***0.1353***	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353	0.1353
34	41	***dst_host_srv_rerror_rate***	0.1200	***0.1155***	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200
Total			**9.7334**	**9.6975**	**9.7319**	**9.7334**	**9.7334**	**9.7334**	**9.7334**	**9.7334**	**9.7334**
Selisih dengan Total Information-Gain fitur sebelum normalisasi				**0.0359**	**0.0014**	-	-	-	-	-	-

Tabel 4.7 yang menyajikan hasil normalisasi dengan metode Log tampak ada 14 fitur yang mengalami perubahan *information-gain* untuk pembulatan hasil normalisasi dengan pembulatan di bawah 7 digit desimal, yaitu: 1) duration, 2) src_bytes, 3) dst_bytes, 4) count, 5) serror_rate, 6) dst_host_same_srv_rate, 7) dst_host_same_src_port_sate, 8) dst_host_srv_diff_host_rate, 9) same_srv_rate, 10) diff_srv_rate, 11) dst_host_serror_rate, 12) dst_host_srv_serror_rate, 13) dst_host_rerror_rate, dan 14) dst_host_srv_rerror_rate.

Ada empat fitur yang mengalami perubahan *information-gain* akibat pembulatan hasil normalisasi pada ketiga metode normalisasi, yaitu: 1) src_bytes, 2) dst_bytes, 3) duration, dan 4) count. Untuk mengetahui lebih jauh tentang kondisi nilai fitur-fitur yang beresiko mengalami perubahan Information-Gain, kami menampilkan data *statistical description* fitur-fitur tersebut pada Tabel 4.8. Terlihat pada Tabel 4.8 bahwa keempat fitur tersebut memiliki standar deviasi yang tinggi, yaitu diatas 100.

Tabel 4.8 Statistic Description dari fitur-fitur yang beresiko mengalami perubahan Information-Gain

	Nama Fitur	Jumlah record	Total nilai	Nilai minimum	Nilai maksimum	Nilai rata-rata	Standar Deviasi
1	**src_bytes**	**125,973**	**5,740,179,316**	**0**	**1,379,963,888**	**45,567.74**	**5,870,331.18**
2	**dst_bytes**	**125,973**	**2,491,634,381**	**0**	**1,309,937,401**	**19,779.11**	**4,021,269.15**
3	**duration**	**125,973**	**36,172,473**	**0**	**42,908**	**287.14**	**2,604.52**
4	**count**	**125,973**	**10,595,281**	**0**	**511**	**84.11**	**114.51**
5	serror_rate	125,973	35837.37	0	1	0.28	0.45
6	dst_host_same_srv_rate	125,973	65662.38	0	1	0.52	0.45
7	dst_host_srv_serror_rate	125,973	35081.53	0	1	0.28	0.45
8	same_srv_rate	125,973	83259.04	0	1	0.66	0.44
9	dst_host_serror_rate	125,973	35833.33	0	1	0.28	0.44
10	dst_host_srv_rerror_rate	125,973	15146.98	0	1	0.12	0.32
11	dst_host_same_src_port_rate	125,973	18691.73	0	1	0.15	0.31
12	dst_host_rerror_rate	125,973	14969.6	0	1	0.12	0.31
13	diff_srv_rate	125,973	7942.93	0	1	0.06	0.18
14	dst_host_srv_diff_host_rate	125,973	4099.47	0	1	0.03	0.11

Kami juga melakukan analisis yang sama terhadap dataset Kyoto2006++ Ringkasan hasil eksperimen kami sajikan pada Tabel 4.9. Hasil uji coba pada ketiga dataset menunjukkan hasil yang sama yaitu risiko tertinggi pembulatan hasil normalisasi ada pada metode Min-max, diikuti oleh Z-score dan Log.

Tabel 4.9 Perbedaan nilai total Information-Gain sebelum proses normalisasi dan sesudah pembulatan hasil normalisasi dengan menggunakan 2 s/d 9 digit desimal

Dataset	Metode Normalisasi	Perbedaan nilai total Information-Gain fitur setelah hasil normalisasi berdasar digit desimal yang digunakan							
		2 digit	3 digit	4 digit	5 digit	6 digit	7 digit	8 digit	9 digit
NSL-KDD	Min-Max	0.0530	0.0477	0.0375	0.0295	0.0185	0.0075	0.0025	-
	Z-Score	0.0414	0.0314	0.0152	0.0069	0.0023	-	-	-
	Log	0.0011	0.00004	-	-	-	-	-	-
KYOTO	Min-Max	0.0571	0.0557	0.0455	0.0386	0.0276	0.0109	0.0045	0.0001
	Z-Score	0.0448	0.0328	0.0260	0.0102	0.0033	0.0002	-	-
	Log	0.0007	0.00002	-	-	-	-	-	-

Dari analisis di atas, kami mengambil empat kesimpulan terkait penggunaan metode normalisasi pada IDS, yaitu:

1. Kinerja dari pengklasifikasi SVM dan k-NN dalam hal ketepatan dan kecepatan dipengaruhi oleh jenis metode normalisasi dan jumlah digit yang digunakan pada proses pembulatan hasil normalisasi.
2. Penggunaan digit desimal yang berbeda untuk pembulatan hasil normalisasi dapat mempengaruhi urutan fitur hasil seleksi
3. Penggunaan metode normalisasi Log pada SVM dan k-NN menghasilkan kecepatan, *accuracy*, *sensitivity*, dan *specificity* yang lebih baik dibanding metode normalisasi Min-max dan Z-score.
4. Penggunaan metode Log dengan 6 digit desimal untuk pembulatan hasil normalisasi dapat digunakan pada dataset NSL-KDD dan Kyoto2006++ tanpa merubah nilai *mutual information* dari fitur-fitur dataset.

4.2. Seleksi Fitur

Pada penelitian ini diajukan dua metode seleksi fitur yang mendukung kondisi *imbalanced-class*, yaitu MRIGFS dan RWIGFS. Dan untuk mengetahui sejauh mana kinerja dari metode yang diajukan, akan dibandingkan dengan metode RIGFS original.

Hasil tahapan pertama seleksi fitur metode MRIGFS dan RWIGFS yang berupa urutan fitur berdasar metode filter pada dataset NSL-KDD disajikan pada Tabel 4.10. Selanjutnya pada tahapan kedua yang menerapkan metode *wrapper* dengan pengklasifikasi SVM standar, hasilnya disajikan pada Gambar 4.8-4.12.

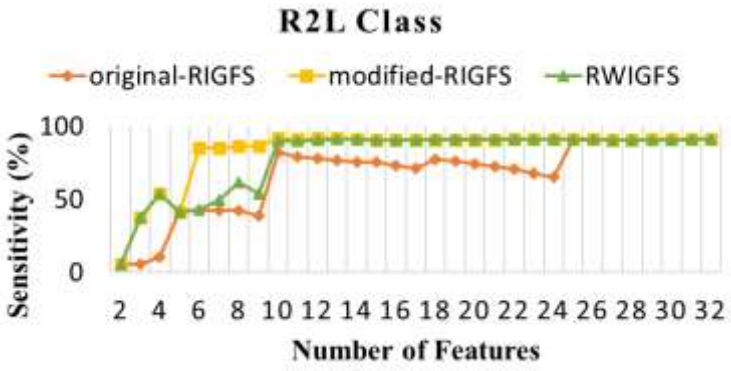
Tabel 4.10 Daftar 40 fitur urutan teratas hasil seleksi fitur

Rank	Original-RIGFS	MRIGFS	RWIGFS
1	duration	src_bytes	src_bytes
2	protocol_type	service	service
3	service	dst_bytes	dst_bytes
4	flag	dst_host_srv_count	dst_host_srv_count
5	src_bytes	hot	hot
6	dst_bytes	dst_host_same_src_port_rate	count
7	land	dst_host_srv_diff_host_rate	dst_host_diff_srv_rate
8	wrong_fragment	srv_count	duration
9	urgent	count	flag
10	hot	duration	dst_host_count
11	num_failed_logins	dst_host_count	diff_srv_rate
12	logged_in	is_guest_login	srv_count
13	num_compromised	srv_diff_host_rate	dst_host_same_src_port_rate
14	root_shell	dst_host_diff_srv_rate	same_srv_rate
15	su_attempted	dst_host_rerror_rate	dst_host_serror_rate
16	num_root	protocol_type	dst_host_srv_diff_host_rate
17	num_file_creations	dst_host_srv_serror_rate	dst_host_same_srv_rate
18	num_shells	dst_host_same_srv_rate	logged_in
19	num_access_files	dst_host_serror_rate	root_shell
20	is_host_login	num_failed_logins	serror_rate
21	is_guest_login	flag	dst_host_srv_serror_rate
22	count	logged_in	srv_serror_rate
23	srv_count	root_shell	num_file_creations
24	serror_rate	dst_host_srv_rerror_rate	num_compromised
25	srv_serror_rate	num_file_creations	srv_diff_host_rate
26	rerror_rate	num_compromised	dst_host_srv_rerror_rate
27	srv_rerror_rate	diff_srv_rate	dst_host_rerror_rate
28	same_srv_rate	same_srv_rate	is_guest_login
29	diff_srv_rate	num_root	protocol_type
30	srv_diff_host_rate	num_shells	rerror_rate
31	dst_host_count	srv_serror_rate	num_root
32	dst_host_srv_count	urgent	srv_rerror_rate
33	dst_host_same_srv_rate	srv_rerror_rate	num_shells
34	dst_host_diff_srv_rate	rerror_rate	num_failed_logins
35	dst_host_same_src_port_rate	wrong_fragment	urgent
36	dst_host_srv_diff_host_rate	land	wrong_fragment
37	dst_host_serror_rate	num_access_files	num_access_files
38	dst_host_srv_serror_rate	serror_rate	su_attempted
39	dst_host_rerror_rate	su_attempted	land
40	dst_host_srv_rerror_rate	is_host_login	is_host_login

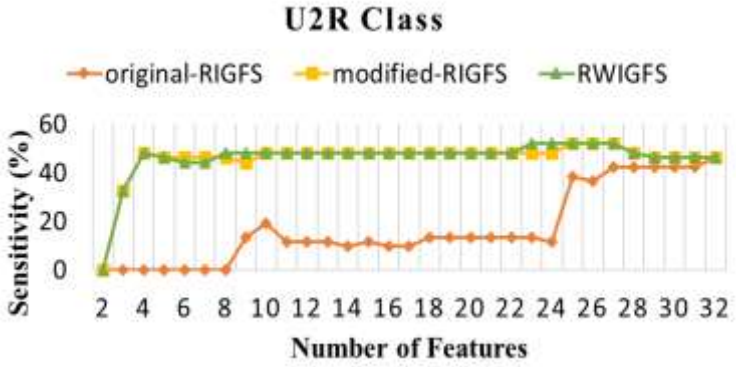
Tabel 4.10 memperlihatkan bahwa ketiga metode seleksi fitur menghasilkan urutan fitur yang berbeda, namun ada persamaan pada lima urutan fitur teratas yang dihasilkan oleh RIGFS dan RWIGFS.

Gambar 4.8 - 4.12 menunjukkan perbandingan sensitivitas pada masing-masing kelas serangan, yaitu R2L, U2R, Normal, Probe, dan DoS. Terlihat pada

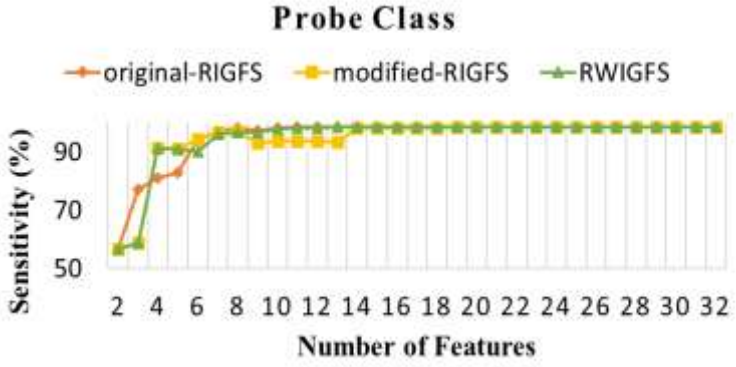
Gambar 4.8 dan 4.9 bahwa metode MRIGFS dan RWIGFS menyebabkan peningkatan yang signifikan dalam pendeteksian kelas minoritas. Kedua metode ini lebih unggul di kelas R2L dan U2R, terutama untuk jumlah fitur kurang dari 25. Ini terjadi karena kedua metode tersebut dapat menempatkan fitur-fitur yang paling berpengaruh untuk deteksi R2L dan U2R di 12 fitur peringkat teratas.



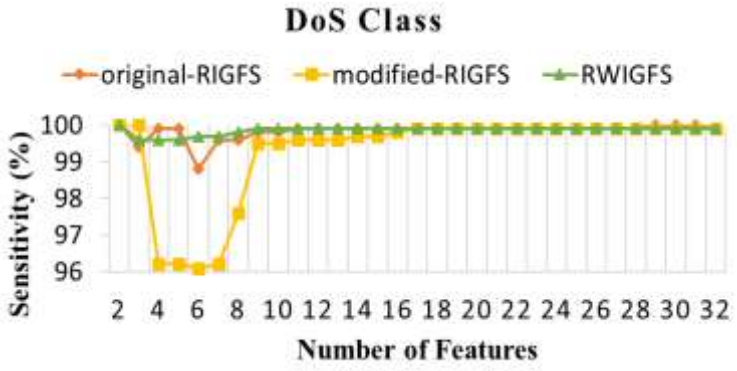
Gambar 4.8 Perbandingan sensitivity pada kelas R2L



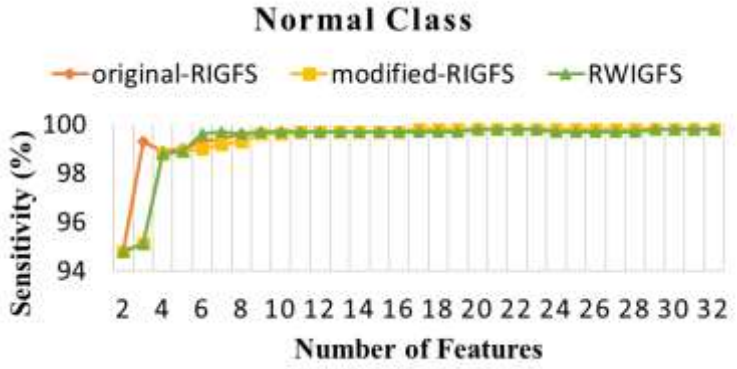
Gambar 4.9 Perbandingan sensitivity pada kelas U2R



Gambar 4.10 Perbandingan sensitivity pada kelas Probe



Gambar 4.11 Perbandingan sensitivity pada kelas DoS



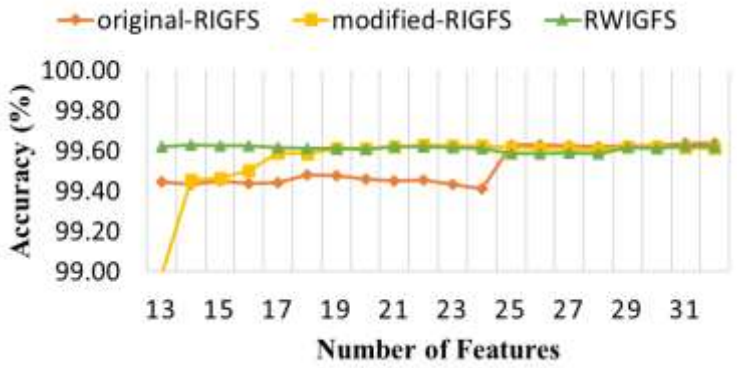
Gambar 4.12 Perbandingan sensitivity pada kelas Normal

Gambar 4.10 - 4.12 menunjukkan bahwa MRIGFS dapat menjaga deteksi kinerja kelas mayoritas tetap tinggi, terutama jika menggunakan jumlah fitur lebih dari 16. Kondisi yang lebih baik ditunjukkan oleh RWIGFS yang dapat menjaga deteksi kinerja kelas mayoritas tetap tinggi dengan jumlah fitur lebih dari 9.

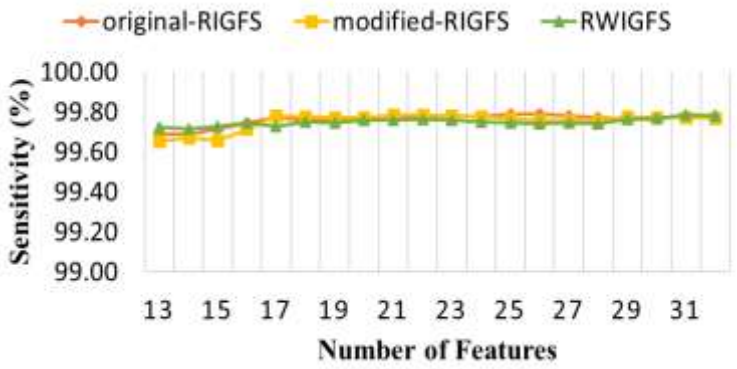
Gambar 4.12 menunjukkan penurunan kinerja deteksi di kelas Normal hanya terjadi ketika menggunakan kurang dari 10 fitur peringkat teratas dari metode MRIGFS, sedangkan untuk RWIGFS penurunan terjadi ketika menggunakan kurang dari 6 fitur peringkat teratas. Gambar 4.10 menunjukkan penurunan kinerja deteksi di kelas Probe yang terjadi dalam penggunaan dataset yang jumlah fiturnya adalah 3, 9, 10, 11, 12, dan 13 untuk metode MRIGFS. Sedangkan di kelas DoS, pada Gambar 4.11, untuk metode MRIGFS penurunan kinerja deteksi terjadi dalam penggunaan dataset yang jumlah fiturnya kurang dari 17.

Pengamatan kinerja deteksi keseluruhan kelas dapat dilakukan menggunakan Gambar 4.13 - 4.17. Gambar-gambar tersebut secara berurutan menunjukkan perbandingan tiga metode pemilihan fitur dalam satuan ukur *accuracy*, *sensitivity*, *specificity*, *G-mean*, dan CPI.

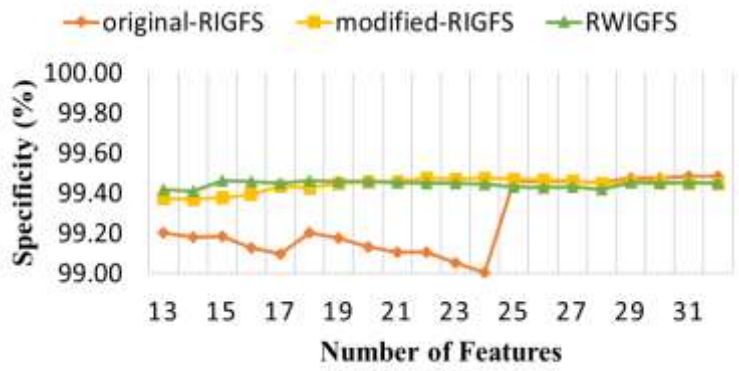
Terlihat terjadi peningkatan pada grafik *accuracy*, grafik *sensitivity*, grafik *specificity*, dan grafik *G-mean* ketika menggunakan metode MRIGFS, dari 13 hingga 17 fitur peringkat teratas. Ketika menggunakan lebih dari 17 fitur peringkat teratas, grafik cenderung stabil. *Accuracy* stabil pada kisaran 99,6%, *sensitivity* stabil pada kisaran 99,7%, dan *G-mean* stabil pada kisaran 99,6%. Kondisi lebih baik ditunjukkan oleh metode RWIGFS yang mencapai kestabilan dengan jumlah fitur yang lebih rendah yaitu 13 fitur peringkat teratas.



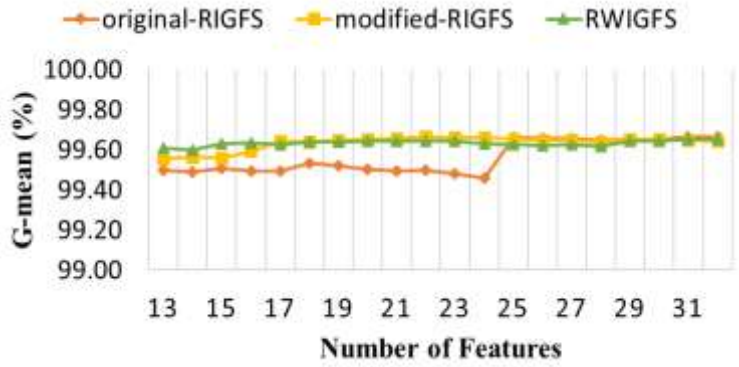
Gambar 4.13 Perbandingan *accuracy* keseluruhan kelas



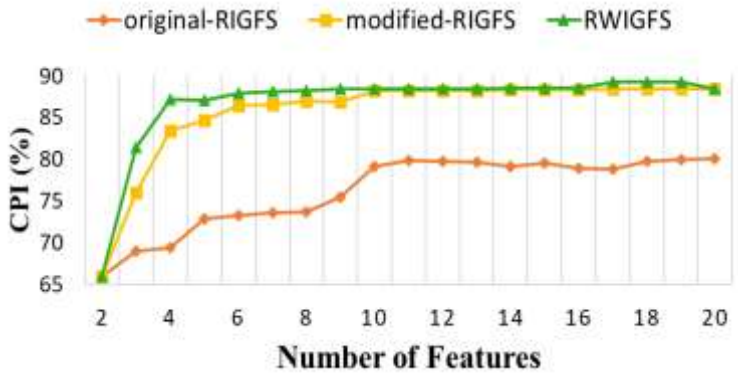
Gambar 4.14 Perbandingan *sensitivity* keseluruhan kelas



Gambar 4.15 Perbandingan *specificity* keseluruhan kelas



Gambar 4.16 Perbandingan G-mean keseluruhan kelas



Gambar 4.17 Perbandingan nilai CPI

Dari pengamatan grafik indek CPI pada gambar 4.17 diketahui bahwa pada rentang jumlah fitur dari 2 sampai 20, MRIGFS yang menghasilkan nilai CPI tertinggi dengan menggunakan 19 fitur peringkat teratas. Sedangkan RWIGFS menghasilkan nilai CPI tertinggi dengan menggunakan 18 fitur peringkat teratas. Pada implementasi dengan menggunakan pengklasifikasi SVM, metode seleksi

fitur MRIGFS dan RWIGFS menghasilkan subset dengan lebih sedikit fitur, mengurangi waktu pelatihan dan meningkatkan kinerja deteksi di semua kelas serangan dibanding dengan metode original-RIGFS. Pada uji coba dengan dataset NSL-KDD, metode MRIGFS menghasilkan fitur-fitur terbaik yang terdiri dari 19 fitur peringkat teratas, sedangkan metode RWIGFS menghasilkan fitur-fitur terbaik yang terdiri dari 18 fitur peringkat teratas.

4.3. IDS berbasis Centroid-based Classification

Bagian ini menyajikan hasil uji coba pengklasifikasi I (L-SCANN) dengan dataset NSL-KDD dan KYOTO 2006++ menggunakan metode normalisasi Log dan metode seleksi fitur MRIGFS dan RWIGFS. Uji coba pelatihan dan pengujian dilakukan dengan metoda 10-*fold cross validation*.

Tabel 4.11 Kinerja L-SCANN pada dataset 19 fitur hasil seleksi MRIGFS

	Class	**k=1**	**k=3**	**k=5**	**k=7**	**k=9**	**k=11**	**k=13**	**k=15**	**k=17**	**k=19**
Accuracy	Normal	93.50%	96.35%	96.73%	96.95%	97.05%	97.11%	97.14%	97.16%	97.17%	97.17%
	DoS	95.96%	96.64%	96.72%	96.74%	96.73%	96.76%	96.75%	96.74%	96.73%	96.73%
	R2L	67.24%	67.24%	65.23%	66.63%	65.73%	65.13%	64.92%	65.13%	64.82%	64.02%
	Probe	91.14%	91.58%	92.12%	92.17%	92.32%	92.05%	92.06%	92.09%	92.23%	92.20%
	U2R	38.46%	38.46%	38.46%	38.46%	38.46%	38.46%	38.46%	38.46%	38.46%	38.46%
	Semua	93.95%	95.76%	96.03%	96.17%	96.23%	96.24%	96.25%	96.26%	96.27%	96.26%
Sensitivity		96.11%	96.10%	96.18%	96.21%	96.21%	96.18%	96.17%	96.16%	96.17%	96.16%
Specificity		93.50%	96.35%	96.73%	96.95%	97.05%	97.11%	97.14%	97.16%	97.17%	97.17%
FPR		6.50%	3.65%	3.27%	3.05%	2.95%	2.89%	2.86%	2.84%	2.83%	2.83%
FNR		3.89%	3.90%	3.82%	3.79%	3.79%	3.82%	3.83%	3.84%	3.83%	3.84%
TP		56352	56346	56393	56406	56410	56388	56385	56381	56382	56380
TN		62964	64884	65142	65288	65356	65397	65419	65431	65438	65439
FP		4379	2459	2201	2055	1987	1946	1924	1912	1905	1904
FN		2278	2284	2237	2224	2220	2242	2245	2249	2248	2250
CPI		0.5474	0.5272	0.7449	0.8090	**0.8276**	0.7219	0.7076	0.6890	0.6935	0.6833

a) Hasil uji coba dengan dataset NSL-KDD

Hasil uji coba L-SCANN dengan subset data hasil seleksi fitur menggunakan MRIGFS kami sajikan pada Tabel 4.11 dan 4.13. Tabel 4.11 menunjukkan bahwa nilai optimal parameter k untuk subset data ini adalah 9. Hal

ini ditunjukkan nilai CPI tertinggi yaitu 0.8276 diperoleh ketika digunakan nilai k=9. Tabel 4.12 menyajikan *confusion-matrix* yang dihasilkan dari kombinasi MRIGFS dan L-SCANN dengan k=9.

Pada Tabel 4.13 kami sajikan hasil uji coba kombinasi RWIGFS dengan L-SCANN. Nilai optimal parameter k yang didapatkan dari hasil uji coba adalah 5, dengan nilai CPI= 0.7291. *Confusion-matrix* yang dihasilkan untuk k=5, kami sajikan pada Tabel 4.14.

Tabel 4.12 *Confusion-matrix* dari pengujian L-SCANN dengan k=9 pada dataset 19 fitur hasil MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	66011	258	100	966	8
	DoS	2338	39841	36	3712	0
	R2L	848	0	143	3	1
	Probe	983	183	58	10432	0
	U2R	46	0	1	2	3

Tabel 4.13 Kinerja L-SCANN pada dataset 18 fitur hasil seleksi RWIGFS

	Class	**k=1**	**k=3**	**k=5**	**k=7**	**k=9**	**k=11**	**k=13**	**k=15**	**k=17**	**k=19**
Accuracy	Normal	94.96%	97.10%	97.41%	97.43%	97.50%	97.54%	97.57%	97.58%	97.59%	97.60%
	DoS	94.97%	95.12%	96.27%	96.23%	96.20%	96.18%	96.15%	96.15%	96.13%	96.11%
	R2L	4.32%	3.52%	3.22%	3.32%	3.22%	3.12%	3.12%	3.12%	3.12%	3.02%
	Probe	92.29%	93.68%	93.94%	94.05%	94.11%	94.12%	94.11%	94.10%	94.11%	94.08%
	U2R	21.15%	21.15%	21.15%	21.15%	21.15%	21.15%	21.15%	21.15%	21.15%	21.15%
	Semua	93.97%	95.29%	95.90%	95.91%	95.93%	95.95%	95.96%	95.96%	95.96%	95.95%
Sensitivity		94.76%	94.87%	94.90%	94.88%	94.87%	94.86%	94.83%	94.82%	94.81%	94.78%
Specificity		94.96%	97.10%	97.41%	97.43%	97.50%	97.54%	97.57%	97.58%	97.59%	97.60%
FPR		5.04%	2.90%	2.59%	2.57%	2.50%	2.46%	2.43%	2.42%	2.41%	2.40%
FNR		5.24%	5.13%	5.10%	5.12%	5.13%	5.14%	5.17%	5.18%	5.19%	5.22%
TP		55558	55621	55642	55630	55625	55616	55599	55593	55585	55570
TN		63952	65388	65597	65615	65658	65685	65705	65714	65718	65724
FP		3391	1955	1746	1728	1685	1658	1638	1629	1625	1619
FN		3072	3009	2988	3000	3005	3014	3031	3037	3045	3060
CPI		0.4225	0.6504	**0.7291**	0.6854	0.6672	0.6346	0.5736	0.5522	0.5239	0.4710

Tabel 4.14 *Confusion-matrix* dari pengujian L-SCANN dengan k=5 pada dataset 18 fitur hasil RWIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	65597	965	52	697	32
	DoS	1525	44212	11	177	2
	R2L	863	6	32	90	4
	Probe	569	114	23	10950	0
	U2R	31	0	6	4	11

Hasil uji coba diatas menunjukkan bahwa implementasi L-SCANN dengan metode normalisasi Log dan kedua metode seleksi fitur dapat mendeteksi semua kelas serangan sehingga kondisi *completeness* terpenuhi. Secara berturutan *sensitivity*, *specificity*, *accuracy* keseluruhan, *accuracy* kelas Normal, *accuracy* kelas DoS, *accuracy* kelas R2L, *accuracy* kelas Probe, dan *accuracy* kelas U2R yang dihasilkan L-SCANN dengan metode seleksi fitur MRIGFS adalah 96.21%, 97.05%, 96.23%, 97.05%, 96.73%, 65.73%, 92.32%, dan 38.46%. Dengan urutan yang sama, kinerja yang dihasilkan L-SCANN dengan metode seleksi fitur RWIGFS adalah 94.90%, 97.41%, 95.90%, 97.41%, 96.27%, 3.22%, 94.05%, dan 21.15%. Metode seleksi fitur MRIGFS lebih baik dibanding RWIGFS pada *sensitivity*, *accuracy* keseluruhan, *accuracy* pada kelas DoS, *accuracy* pada kelas R2L, dan *accuracy* pada kelas U2R. Sedangkan pada *specificity*, *accuracy* kelas Normal dan *accuracy* pada kelas Probe, metode seleksi fitur RWIGFS lebih baik dibanding MRIGFS. Metode seleksi fitur MRIGFS juga menghasilkan jumlah FN (serangan yang dianggap sebagai aktivitas normal) yang lebih sedikit dibanding yang dihasilkan oleh RWIGFS, 2220 dibanding 2988. Hal tersebut menunjukkan bahwa metode seleksi fitur MRIGFS lebih sesuai untuk diterapkan pada L-SCANN.

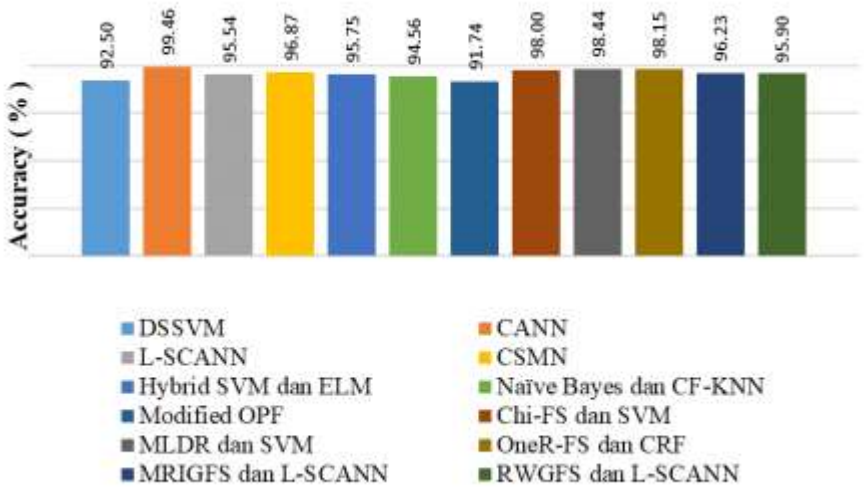
Selanjutnya kami akan membandingkan kinerja model IDS tersebut dengan model IDS yang ada sebelumnya, data perbandingan kami sajikan pada Tabel 4.15, Gambar 4.18, dan Gambar 4.19. Gambar 4.18 menunjukkan perbandingan *accuracy* keseluruhan kelas dari 12 model IDS dalam bentuk grafik diagram batang. Gambar 4.19 menunjukkan perbandingan *accuracy* masing-masing kelas dari 12 model IDS dalam bentuk grafik diagram batang. Dari kiri ke kanan adalah kelas Normal, DoS, R2L, Probe, dan U2R.

Model IDS yang digunakan sebagai pembanding adalah DSSVM (Guo et al., 2014), CANN (Lin et al., 2015), L-SCANN (Ahmad & Muchammad, 2016), CSMN (Muttaqien & Ahmad, 2017), Hybrid SVM and ELM (Al-Yaseen et al., 2017), Naïve Bayes and CF-KNN (Pajouh et al., 2017), Modified OPF (Bostani & Sheikhan, 2017), Chi-FS and multi-class SVM (Sumaiya Thaseen & Aswani Kumar, 2017), MLDR and multi-class SVM (Kumar et al., 2018), dan OneR-FS and CRF (Mahendiran & Appusamy, 2018)

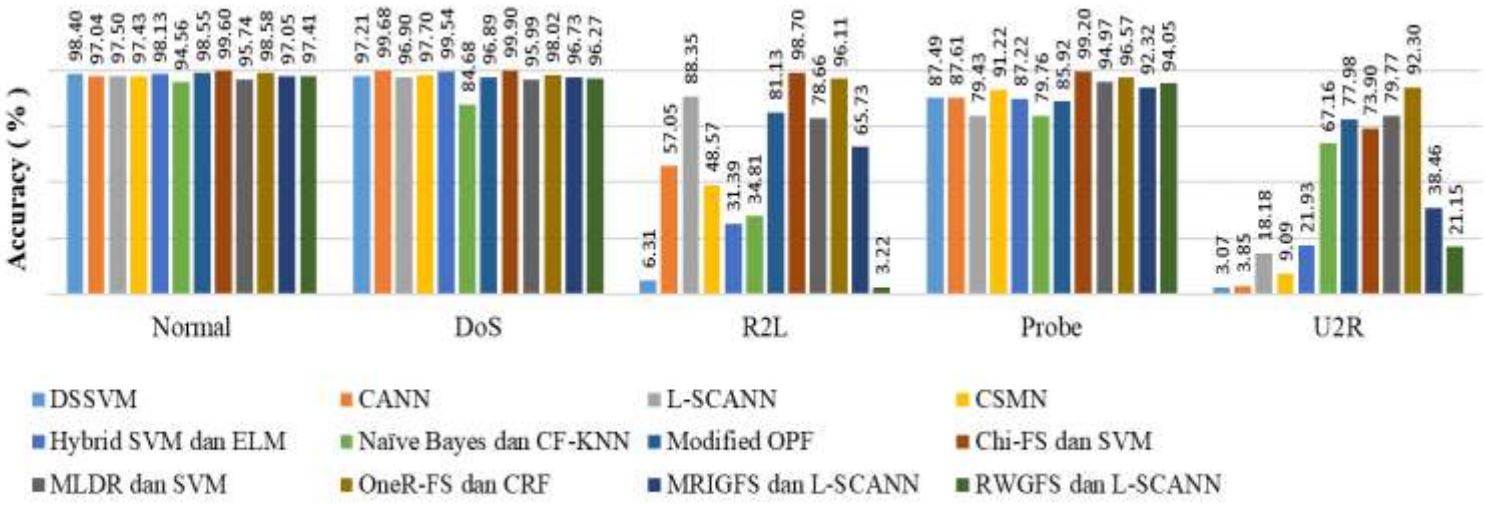
Pertama, kami bandingkan model IDS tersebut (L-SCANN + normalisasi Log + MRIGFS) dengan 4 model IDS yang menggunakan algoritma CBC pada penelitian terdahulu yaitu DSSVM (Guo et al., 2014), CANN (Lin et al., 2015), L-SCANN (Ahmad & Muchammad, 2016), dan CSMN (Muttaqien & Ahmad, 2017) untuk kinerja deteksi terhadap keseluruhan kelas ada pada urutan 3. Untuk *accuracy* masing-masing kelas (Normal, DoS, Probe, R2L, dan U2R) secara berurutan peringkatnya adalah 4, 5, 1, 2, dan 1. Dapat kita lihat bahwa kinerja deteksi pada kelas U2R dan R2L yang merupakan *minority-class*, model yang diusulkan lebih baik dibanding penelitian berbasis CBC sebelumnya. Pada kelas Normal, DoS, dan Probe yang merupakan *majority-class*, *accuracy* nya memang lebih rendah tetapi masih diatas 92%. Hal ini menunjukkan bahwa model yang diusulkan dapat meningkatkan kinerja deteksi pada *minority class*, dan mempertahankan kinerja deteksi pada kelas lainnya tetap tinggi. Kedua, kami bandingkan model IDS tersebut dengan 10 model IDS pembanding yang ada di Tabel 4.15. Untuk kinerja deteksi terhadap keseluruhan kelas, model yang kami usulkan ada pada urutan 6 dari 11 model IDS. Urutan rangking kami berikan di belakang nilai *accuracy*. Untuk *accuracy* masing-masing kelas (Normal, DoS, Probe, R2L, dan U2R) secara berurutan peringkatnya adalah 8, 9, 4, 5, dan 6. Angka di dalam tanda kurung ‘()’ merupakan urutan peringkat.

Tabel 4.15 Perbandingan kinerja dengan model IDS lain

Metode IDS	∑ fitur	Accuracy (%)					
		Overall	Normal	DoS	R2L	Probe	U2R
DSSVM (Guo et al., 2014)	52	92.50	98.40(4)	97.21(6)	6.31(10)	87.49	3.07
CANN (Lin et al., 2015)	19	99.46(1)	97.04	99.68(2)	57.05(6)	87.61	3.85
L-SCANN (Ahmad & Muchammad, 2016)	19	95.54	97.50(6)	96.90(7)	88.35(3)	79.43	18.18
CSMN (Muttaqien & Ahmad, 2017)	19	96.87(5)	97.43(7)	97.70(5)	48.57(7)	91.22	9.09
Hybrid SVM and ELM (Al-Yaseen et al., 2017)	---	95.75	98.13(5)	99.54(3)	31.39(9)	87.22	21.93(7)
Naïve Bayes and CF-KNN (Pajouh et al., 2017)	---	94.56	94.56	84.68	34.81(8)	79.76	67.16(5)
Modified OPF (Bostani & Sheikhan, 2017)	---	91.74	98.55(3)	96.89(8)	81.13(3)	85.92	77.98(3)
Chi-FS and multi-class SVM (Sumaiya Thaseen & Aswani Kumar, 2017)	31	98.00(4)	99.60(1)	99.90(1)	98.70(1)	99.20(1)	73.90(4)
MLDR and multi-class SVM (Kumar et al., 2018)	---	98.44(2)	95.74	95.99	78.66(4)	94.97(3)	79.77(2)
OneR-FS and CRF (Mahendiran & Appusamy, 2018)	24	98.15(3)	98.58(2)	98.02(4)	96.11(2)	96.57(2)	92.30(1)
L-SCANN dengan MRIGFS	**19**	**96.23(6)**	**97.05(8)**	**96.73(9)**	**65.73(5)**	**92.32(5)**	**38.46(6)**
L-SCANN dengan RWIGFS	**18**	**95.90**	**97.41**	**96.27**	**3.22**	**94.05**	**21.15**



Gambar 4.18 Perbandingan *accuracy* keseluruhan dengan model IDS lain



Gambar 4.19 Perbandingan *accuracy* pada masing-masing kelas dengan model IDS lain

b) Hasil uji coba dengan dataset Kyoto 2006++

Hasil uji coba L-SCANN dengan subset 14 fitur dari dataset Kyoto2006++ kami sajikan pada Tabel 4.16. Dari hasil uji coba kami mendapatkan bahwa nilai optimal parameter k untuk subset data ini adalah 7. Hal ini ditunjukkan nilai CPI tertinggi yaitu 0.9585 diperoleh ketika digunakan nilai k=7. Pada Tabel 4.17 kami sajikan hasil uji coba L-SCANN dengan subset 7 fitur dari dataset Kyoto2006++. Nilai optimal parameter k yang didapatkan dari hasil uji coba adalah 9, dengan nilai CPI= 0.9258.

Tabel 4.16 Kinerja L-SCANN pada dataset 14 fitur Kyoto2006++

	Class	**k=1**	**k=3**	**k=5**	**k=7**	**k=9**	**k=11**	**k=13**	**k=15**	**k=17**	**k=19**
Accuracy	Normal	97.75%	98.51%	98.61%	98.64%	98.68%	98.70%	98.71%	98.73%	98.75%	98.75%
	Attack	89.20%	91.06%	90.26%	91.46%	91.27%	91.29%	91.27%	91.16%	91.13%	91.10%
	Semua	97.03%	97.89%	97.91%	98.04%	98.07%	98.08%	98.08%	98.10%	98.12%	98.11%
Sensitivity		89.20%	91.06%	90.26%	91.46%	91.27%	91.29%	91.27%	91.16%	91.13%	91.10%
Specificity		97.75%	98.51%	98.61%	98.64%	98.68%	98.70%	98.71%	98.73%	98.75%	98.75%
TP		5139	5246	5200	5269	5258	5259	5258	5252	5250	5248
TN		61837	62321	62381	62404	62431	62440	62444	62458	62473	62472
FP		1426	942	882	859	832	823	819	805	790	791
FN		622	515	561	492	503	502	503	509	511	513
CPI		0.9442	0.9561	0.9536	**0.9585**	0.9580	0.9582	0.9581	0.9579	0.9579	0.9577

Tabel 4.17 Kinerja L-SCANN pada dataset 7 fitur Kyoto2006++

	Class	**k=1**	**k=3**	**k=5**	**k=7**	**k=9**	**k=11**	**k=13**	**k=15**	**k=17**	**k=19**
Accuracy	Normal	98.40%	98.93%	99.06%	99.21%	99.17%	99.18%	99.19%	99.20%	99.21%	99.20%
	Attack	81.86%	82.52%	82.24%	82.17%	82.23%	82.10%	82.00%	81.86%	81.77%	81.67%
	Semua	97.02%	97.56%	97.65%	97.70%	97.76%	97.76%	97.75%	97.76%	97.75%	97.74%
Sensitivity		81.86%	82.52%	82.24%	82.17%	82.23%	82.10%	82.00%	81.86%	81.77%	81.67%
Specificity		98.40%	98.93%	99.06%	99.11%	99.17%	99.18%	99.19%	99.20%	99.21%	99.20%
TP		4716	4754	4738	4734	4737	4730	4724	4716	4711	4705
TN		62248	62588	62667	62702	62741	62747	62749	62759	62763	62758
FP		1015	675	596	561	522	516	514	504	500	505
FN		1045	1007	1023	1027	1024	1031	1037	1045	1050	1056
CPI		0.9196	0.9255	0.9252	0.9252	**0.9258**	0.9254	0.9250	0.9246	0.9243	0.9238

Hasil uji coba diatas menunjukkan bahwa implementasi L-SCANN dengan metode normalisasi Log dan dua dataset (7 fitur dan 14 fitur) dari

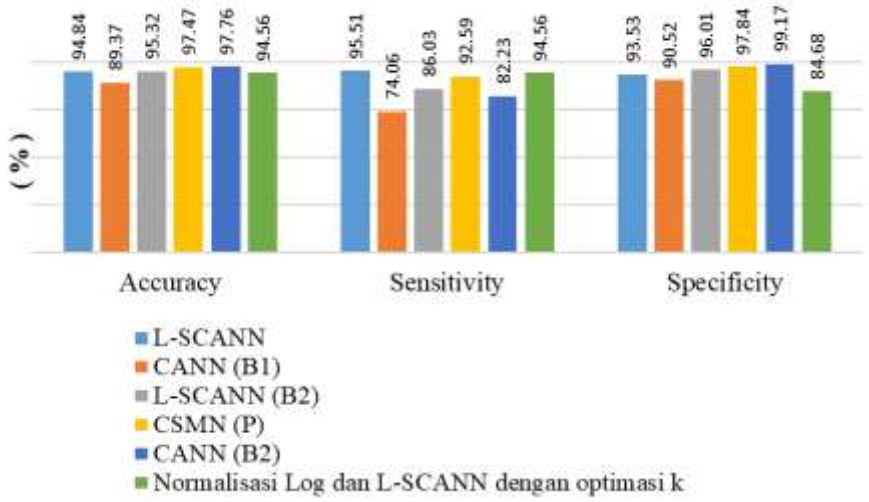
Kyoto2006++ menghasilkan kinerja yang tinggi. Secara berturutan *accuracy*, *sensitivity*, *specificity*, dan FN yang dihasilkan L-SCANN dengan subset 7 fitur dari dataset Kyoto2006++ adalah 97.76%, 82.23%, 99.17%, dan 1024. Dengan urutan yang sama, kinerja yang dihasilkan L-SCANN dengan 98.04%, 91.46%, 98.64%, dan 492. Kinerja L-SCANN dengan subset 14 fitur menghasilkan *accuracy* dan *sensitivity* yang lebih baik dibanding penggunaan subset 7 fitur, namun pada *specificity* lebih rendah. Implementasi L-SCANN dengan subset 14 fitur juga menghasilkan jumlah FN (serangan yang dianggap sebagai aktivitas normal) yang lebih sedikit dibanding yang dihasilkan oleh subset 7 fitur, 492 dibanding 1024. Hal tersebut menunjukkan bahwa subset 14 fitur dari Kyoto2006++ menghasilkan kinerja yang lebih tinggi dibanding penggunaan subset 7 fitur. Selanjutnya kami membandingkan kinerja model IDS yang diuji dengan kinerja model IDS terdahulu yang menggunakan IDS berbasis CBC dan dataset Kyoto2006++. Data perbandingan kami sajikan pada Tabel 4.18, Gambar 4.20, dan Gambar 4.21.

Tabel 4.18 Perbandingan Kinerja dengan model IDS lain

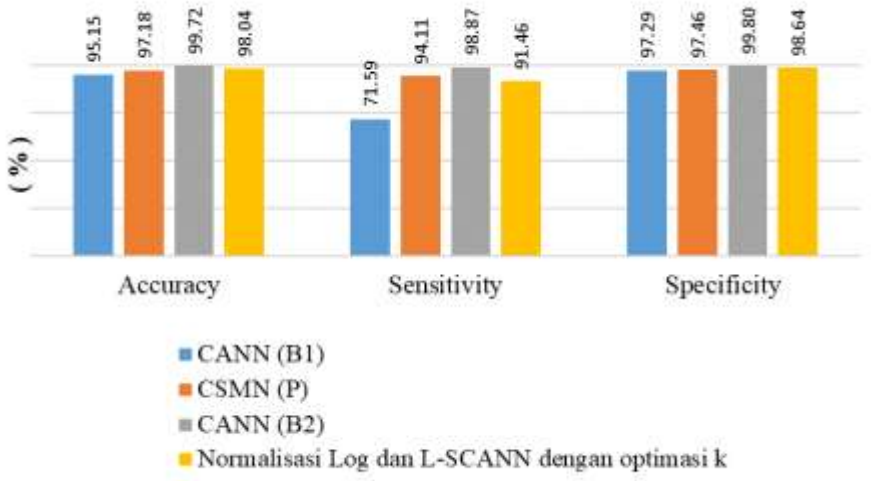
Metode IDS	∑ fitur	Accuracy (%)	Sensitivity (%)	Specificity (%)
L-SCANN (Ahmad & Muchammad, 2016)	7	94.84	95.51	93.53
CANN (B1) (Muttaqien & Ahmad, 2017)	7	89.37	74.06	90.52
L-SCANN (B2) (Muttaqien & Ahmad, 2017)	7	95.32	86.03	96.01
CSMN (P) (Muttaqien & Ahmad, 2017)	7	97.47	92.59	97.84
L-SCANN 7 fitur + log + optimasi k=9	**7**	**97.76(1)**	**82.23(4)**	**99.17(1)**
CANN (B1) (Muttaqien & Ahmad, 2017)	14	95.15	71.59	97.29
L-SCANN (B2) (Muttaqien & Ahmad, 2017)	14	97.18	94.11	97.46
CSMN (P) (Muttaqien & Ahmad, 2017)	14	99.72	98.87	99.80
L-SCANN 14 fitur + log + optimasi k=7	**14**	**98.04(2)**	**91.46(3)**	**98.64(2)**

Tabel 4.18 menunjukkan bahwa pada penggunaan dataset dengan 7 fitur, kinerja model yang diuji dibandingkan dengan hasil 4 penelitian IDS berbasis CBC yang lain memiliki peringkat *accuracy*, *sensitivity*, dan *specificity* secara berturutan adalah 1, 4, dan 1. Sedangkan untuk penggunaan dataset dengan 9 fitur,

dibandingkan dengan hasil 3 penelitian IDS berbasis CBC yang lain, peringkat *accuracy*, *sensitivity*, dan *specificity*-nya secara berturutan adalah 2, 3, dan 2.



Gambar 4.20 Perbandingan dengan model IDS lain untuk dataset Kyoto dengan 7 fitur



Gambar 4.21 Perbandingan dengan model IDS lain untuk dataset Kyoto dengan 14 fitur

Hasil uji coba pada dataset NSL-KDD menunjukkan bahwa implementasi metode normalisasi log dan metode seleksi fitur yang diajukan dapat meningkatkan

kinerja deteksi algoritma L-SCANN pada keseluruhan kelas dan mampu meningkatkan deteksi pada kelas minoritas dan mempertahankan kinerja deteksi pada kelas lainnya tetap tinggi (diatas 92%). Sedangkan hasil uji coba pada dataset Kyoto2006++ menunjukkan bahwa kinerja L-SCANN meningkat dalam hal *accuracy* dan *specificity*, tetapi *sensitivity* –nya sedikit menurun.

4.4. IDS berbasis SVM dengan optimasi parameter kernel

Bagian ini menyajikan hasil eksperimen dari pengklasifikasi II, yaitu SVM dengan optimasi kernel (SVM-OP) dengan dataset NSL-KDD dan KYOTO 2006++ metode normalisasi Log dan metode seleksi fitur MRIGFS dan RWIGFS. Kernel yang digunakan adalah RBF dan parameter yang dioptimasi adalah parameter C dan gamma. Uji coba pelatihan dan pengujian pada dataset Kyoto2006++ dilakukan dengan metoda 10-*fold cross validation.* Sedangkan pada dataset NSL-KDD, uji coba dilakukan dengan membagi 70% untuk data pengujian dan 30% untuk data pelatihan.

a) Hasil uji coba dengan dataset NSL-KDD

Pada Tabel 4.19 kami sajikan hasil uji coba SVM-OP dengan data pengujian dari dataset NSL-KDD data hasil seleksi fitur menggunakan RWIGFS dan MRIGFS. Seperti yang sudah dibahas pada bagian 3.6, nilai pasangan parameter C dan gamma yang digunakan untuk MRIGFS adalah 398.107170 dan 0.038729, sedangkan untuk RWIGFS adalah 63.095734 dan 0.167210.

Confusion-matrix yang dihasilkan dari pelatihan dan pengujian SVM-OP menggunakan subset data hasil seleksi fitur dengan MRIGFS disajikan pada Tabel 4.20 dan Tabel 4.21. Sedangkan *confusion-matrix* yang dihasilkan dari pelatihan dan pengujian SVM-OP menggunakan subset data hasil seleksi fitur dengan RWIGFS disajikan pada Tabel 4.22 dan Tabel 4.23.

Tabel 4.19 Kinerja SVM-OP dengan RWIGFS dan MRIGFS pada data pengujian dari dataset NSL-KDD

Kinerja		MRIGFS	RWIGFS
Accuracy	Normal	99.79%	99.82%
	DoS	99.92%	99.96%
	R2L	91.63%	93.40%
	Probe	99.29%	99.50%
	U2R	57.50%	71.43%
	Semua	99.71%	99.78%
Sensitivity		99.63%	99.77%
Specificity		99.79%	99.82%
FPR		0.21%	0.18%
FNR		0.37%	0.23%
TP		40901	40886
TN		47023	47101
FP		99	85
FN		153	96

Tabel 4.20 *Confusion-matrix* dari pelatihan SVM dengan 19 fitur dari MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	20183	4	13	18	3
	DoS	5	13731	0	1	0
	R2L	20	0	267	0	3
	Probe	21	3	0	3507	0
	U2R	4	0	2	0	6

Tabel 4.21 *Confusion-matrix* dari pengujian SVM dengan 19 fitur dari MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	47023	14	37	39	9
	DoS	24	32165	0	1	0
	R2L	56	1	646	0	2
	Probe	58	0	0	8067	0
	U2R	15	1	1	0	23

Tabel 4.22 *Confusion-matrix* dari pelatihan SVM dengan 18 fitur dari RWIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	20125	8	8	13	3
	DoS	2	13811	0	0	0
	R2L	21	0	307	0	0
	Probe	14	0	0	3465	0
	U2R	4	0	2	1	10

Tabel 4.23 *Confusion-matrix* dari pengujian SVM dengan 18 fitur dari RWIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	47101	13	30	39	3
	DoS	9	32102	0	3	0
	R2L	40	0	623	2	2
	Probe	40	1	0	8136	0
	U2R	7	0	2	1	25

Hasil uji coba diatas menunjukkan bahwa implementasi SVM-OP dengan metode normalisasi Log dan kedua metode seleksi fitur yang diusulkan dapat mendeteksi semua kelas serangan sehingga kondisi *completeness* terpenuhi. Secara berturutan *sensitivity*, *specificity*, *accuracy* keseluruhan, *accuracy* kelas Normal, *accuracy* kelas DoS, *accuracy* kelas R2L, *accuracy* kelas Probe, dan *accuracy* kelas U2R yang dihasilkan SVM-OP dengan metode seleksi fitur MRIGFS adalah 99.63%, 99.79%, 99.71%, 99.79%, 99.92%, 91.63%, 99.29%, dan 57.50%. Dengan urutan yang sama, kinerja yang dihasilkan SVM-OP dengan metode seleksi fitur RWIGFS adalah 99.77%, 99.82%, 99.78%, 99.82%, 99.96%, 93.40%, 99.50%, dan 71.43%. Metode seleksi fitur RWIGFS lebih baik dari MRIGFS pada keseluruhan ukuran kinerja yang diuji. Metode seleksi fitur RWIGFS juga menghasilkan jumlah FN (serangan yang dianggap sebagai aktivitas normal) yang lebih sedikit dibanding yang dihasilkan oleh MRIGFS, 96 dibanding 153. Hal tersebut menunjukkan bahwa metode seleksi fitur RWIGFS lebih sesuai untuk diterapkan pada SVM-OP.

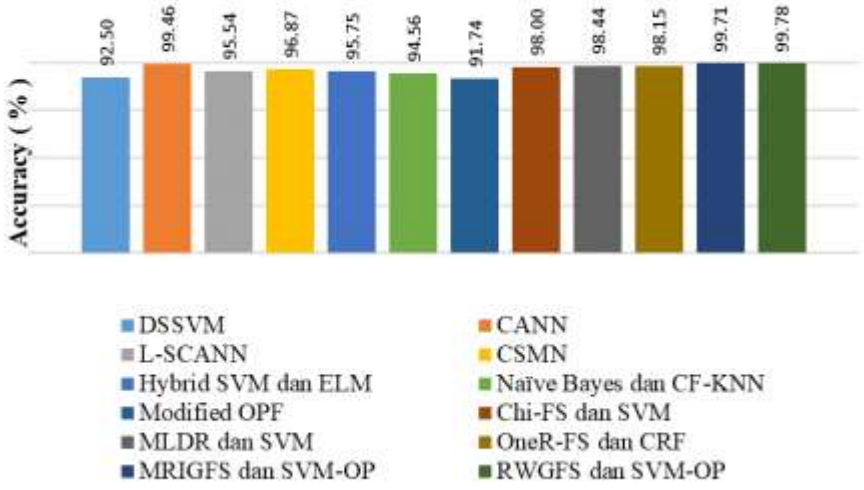
Selanjutnya kami akan membandingkan kinerja kedua model IDS tersebut dengan model IDS yang ada sebelumnya, data perbandingan kami sajikan pada Tabel 4.24, Gambar 4.22, dan Gambar 4.23. Gambar 4.22 menunjukkan perbandingan *accuracy* keseluruhan kelas dari 12 model IDS dalam bentuk grafik diagram batang. Gambar 4.23 menunjukkan perbandingan *accuracy* masing-masing kelas dari 12 model IDS dalam bentuk grafik diagram batang. Dari kiri ke kanan adalah kelas Normal, DoS, R2L, Probe, dan U2R.

Tampak di Tabel 4.24 bahwa kedua model berbasis SVM-OP ini ada pada urutan 1 dan 2 dari 12 model IDS untuk kinerja *accuracy* keseluruhan kelas, *accuracy* kelas Normal, *accuracy* kelas DoS, dan *accuracy* kelas Probe. Untuk

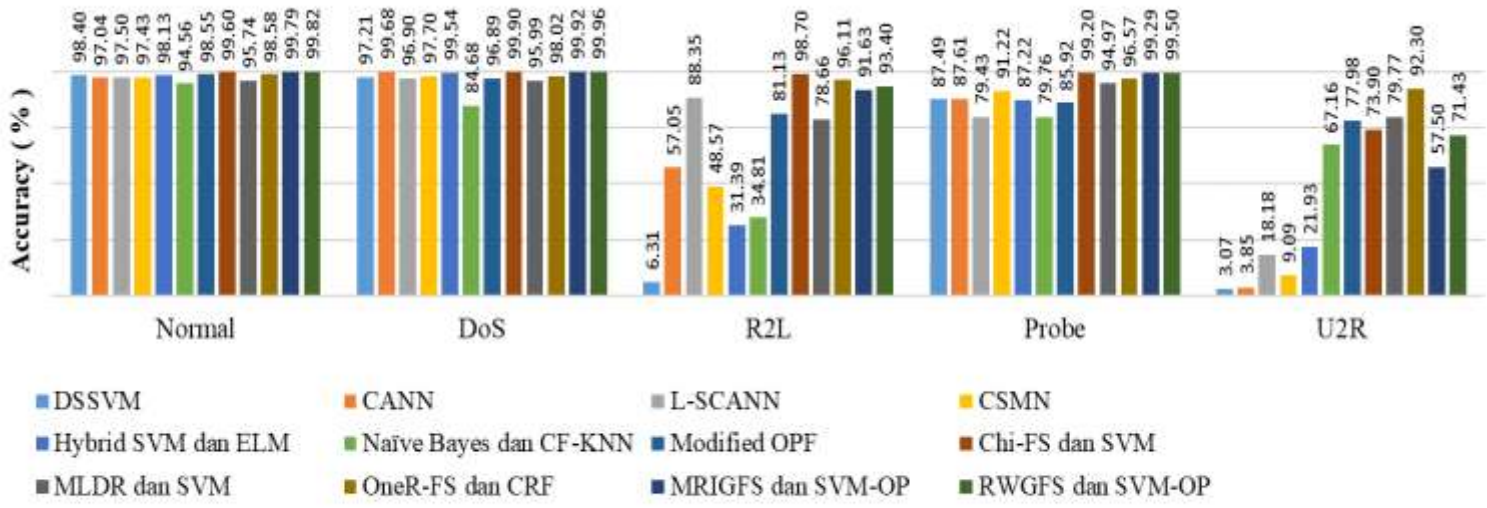
accuracy pada kelas R2L ada pada urutan 3 dan 4, sedangkan *accuracy* pada kelas U2R ada pada urutan 5 dan 7.

Tabel 4.24 Perbandingan kinerja dengan model IDS lain

Metode IDS	∑ fitur	Accuracy (%)					
		Overall	Normal	DoS	R2L	Probe	U2R
DSSVM (Guo et al., 2014)	52	92.50	98.40	97.21	6.31	87.49	3.07
CANN (Lin et al., 2015)	19	99.46	97.04	99.68	57.05	87.61	3.85
L-SCANN (Ahmad & Muchammad, 2016)	19	95.54	97.50	96.90	88.35	79.43	18.18
CSMN (Muttaqien & Ahmad, 2017)	19	96.87	97.43	97.70	48.57	91.22	9.09
Hybrid SVM and ELM (Al-Yaseen et al., 2017)	---	95.75	98.13	99.54	31.39	87.22	21.93
Naïve Bayes and CF-KNN (Pajouh et al., 2017)	---	94.56	94.56	84.68	34.81	79.76	67.16
Modified OPF (Bostani & Sheikhan, 2017)	---	91.74	98.55	96.89	81.13	85.92	77.98(3)
Chi-FS and multi-class SVM (Sumaiya Thaseen & Aswani Kumar, 2017)	31	98.00	99.60	99.90	98.70(1)	99.20	73.90(4)
MLDR and multi-class SVM (Kumar et al., 2018)	---	98.44	95.74	95.99	78.66	94.97	79.77(2)
OneR-FS and CRF (Mahendiran & Appusamy, 2018)	24	98.15	98.58	98.02	96.11(2)	96.57	92.30(1)
SVM-OP dengan MRIGFS	**19**	**99.71(2)**	**99.79(2)**	**99.92(2)**	**91.63(4)**	**99.29(2)**	**57.50(7)**
SVM-OP dengan RWIGFS	**18**	**99.78(1)**	**99.82(1)**	**99.96(1)**	**93.40(3)**	**99.50(1)**	**71.43(5)**



Gambar 4.22 Perbandingan *accuracy* keseluruhan dengan model IDS lain



Gambar 4.23 Perbandingan *accuracy* pada masing-masing kelas dengan model IDS lain

b) Hasil uji coba dengan dataset Kyoto 2006++

Hasil optimasi parameter kernel RBF dengan subset data 14 fitur dari dataset Kyoto2006++ menghasilkan pasangan nilai optimal untuk parameter C=63.095734=$10^{1.8}$ dan gamma=0.038729=$10^{-1.4}$. Pada Tabel 4.25 kami sajikan hasil uji coba SVM-OP dengan nilai parameter optimal pada data pengujian Kyoto2006++ 14 fitur. *Confusion-matrix* yang dihasilkan dari uji coba tersebut kami sajikan pada Tabel 4.26.

Tabel 4.25 Kinerja SVM dengan C=63.095734 dan gamma=0.038729 pada dataset Kyoto2006++ 14 fitur

Kinerja		**nilai**
Accuracy	Normal	99.86%
	Attack	99.69%
	Semua	99.85%
Sensitivity		99.69%
Specificity		99.86%
TP		5743
TN		63175
FP		88
FN		18

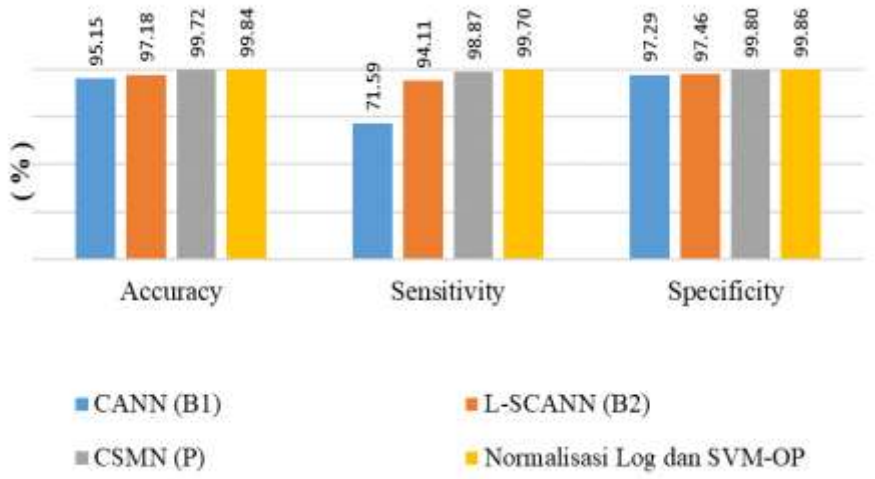
Tabel 4.26 *Confusion-matrix* dari pengujian SVM dengan C=63.095734 dan gamma=0.038729 pada dataset Kyoto2006++ 14 fitur

		Prediksi	
		Normal	**Attack**
Aktual	**Normal**	63175	88
	Attack	18	5743

Implementasi algoritma SVM dengan optimasi kernel RBF dan metode normalisasi log pada dataset 14 fitur dari Kyoto2006++ menghasilkan *accuracy*, *sensitivity*, dan *specificity* yang tinggi, diatas 99%. Ketika dibandingkan dengan model IDS yang ada sebelumnya yang menggunakan dataset Kyoto2006++ 7 fitur dan 14 fitur, model IDS ini unggul pada *accuracy*, *sensitivity*, dan *specificity*. Data perbandingan kinerja kami sajikan pada Tabel 4.27.

Tabel 4.27 Perbandingan kinerja dengan model IDS lain

Metode IDS	∑ fitur	Accuracy (%)	Sensitivity (%)	Specificity (%)
L-SCANN (Ahmad & Muchammad, 2016)	7	94.84	95.51	93.53
CANN (B1) (Muttaqien & Ahmad, 2017)	7	89.37	74.06	90.52
L-SCANN (B2) (Muttaqien & Ahmad, 2017)	7	95.32	86.03	96.01
CSMN (P) (Muttaqien & Ahmad, 2017)	7	97.47	92.59	97.84
CANN (B1) (Muttaqien & Ahmad, 2017)	14	95.15	71.59	97.29
L-SCANN (B2) (Muttaqien & Ahmad, 2017)	14	97.18	94.11	97.46
CSMN (P) (Muttaqien & Ahmad, 2017)	14	99.72	98.87	99.80
Normalisasi Log dan SVM-OP	**14**	**99.84(1)**	**99.70(1)**	**99.86(1)**



Gambar 4.24 Perbandingan dengan model IDS lain untuk dataset Kyoto dengan 14 fitur

Implementasi IDS berbasis SVM-OP yang diusulkan pada dataset NSL-KDD dan dataset Kyoto2006++ dapat mencapai kondisi *completeness* karena dapat mendeteksi semua kelas serangan pada dataset yang diuji. Berikut tiga kesimpulan yang dapat kami ambil dari uji coba model IDS berbasis SVM-OP:

1. Pada uji coba SVM-OP pada dataset NSL-KDD, penggunaan metode seleksi fitur RWIGFS pada SVM-OP menghasilkan kinerja lebih baik dibanding jika menggunakan MRIGFS pada keseluruhan ukuran kinerja

yang diuji (*sensitivity*, *specificity*, *accuracy* pada keseluruhan kelas, dan *accuracy* pada masing-masing kelas). Disamping itu juga menghasilkan jumlah *false negative* yang lebih sedikit.

2. Kedua model IDS berbasis SVM-OP ini mempunyai kinerja yang cukup baik jika dibandingkan dengan kinerja dari 10 model IDS pembanding yang menggunakan dataset NSL-KDD. Mereka berada pada urutan 1 dan 2 dari 12 model IDS untuk kinerja *accuracy* keseluruhan kelas, *accuracy* kelas Normal, *accuracy* kelas DoS, dan *accuracy* kelas Probe. Untuk *accuracy* pada kelas R2L ada pada urutan 3 dan 4, sedangkan *accuracy* pada kelas U2R ada pada urutan 6 dan 7.
3. Implementasi algoritma SVM dengan optimasi kernel RBF dan metode normalisasi log pada dataset 14 fitur dari Kyoto2006++ menghasilkan *accuracy*, *sensitivity*, dan *specificity* yang tinggi, diatas 99%. Dibandingkan dengan 7 model IDS yang ada sebelumnya yang menggunakan dataset Kyoto2006++ 7 fitur dan 14 fitur, model IDS ini unggul pada keseluruhan ukuran kinerja yang diuji (*sensitivity, specificity, accuracy*).

4.5. IDS berbasis SVM dengan optimasi bobot kelas

Bagian ini menyajikan hasil eksperimen dari pengklasifikasi III, yaitu *cost learning* SVM dengan optimasi bobot kelas (SVM-OW) dengan dataset NSL-KDD dan KYOTO 2006++. Untuk dataset NSL-KDD, optimasi bobot kelas dilakukan dengan dua cara. Pertama, optimasi bobot dilakukan dengan menggunakan perbandingan bobot berdasarkan jumlah data pada masing-masing kelas serangan (OB-1). Kedua, optimasi bobot dilakukan pada kelas R2L dan U2R yang merupakan *minority class* dengan menggunakan *grid-search* (OB-2). Pencarian bobot optimal dilakukan pada rentang nilai 1 sampai dengan 10. Sedangkan bobot kelas lainnya ditetapkan dengan nilai 1. Hal ini dilakukan dengan tujuan untuk meningkatkan kinerja deteksi pada *minority class* dan mempertahankan kinerja kelas lainnya tetap tinggi. Untuk uji coba pada dataset Kyoto2006++ hanya digunakan optimasi bobot pertama (OB-1). Uji coba pelatihan dan pengujian pada dataset Kyoto2006++ dilakukan dengan metoda 10-*fold cross*

validation. Sedangkan pada dataset NSL-KDD, uji coba dilakukan dengan membagi 70% untuk data pengujian dan 30% untuk data pelatihan.

a) Hasil uji coba dengan dataset NSL-KDD

Pada Tabel 4.28 kami sajikan hasil uji coba SVM-OW dengan data pengujian dari dataset NSL-KDD data hasil seleksi fitur menggunakan RWIGFS dan m-RIGFS. Seperti yang sudah dibahas pada bagian 3.7, nilai komposisi bobot kelas (Normal:DoS:R2L:Probe:U2R) melalui pendekatan perbandingan bobot berdasarkan jumlah data pada masing-masing kelas serangan (OB-1) adalah (2:2:101:9:1923). Melalui pendekatan OB-2 atau menggunakan *grid-search*, nilai optimal komposisi bobot kelas yang didapat untuk subset data hasil seleksi fitur menggunakan MRIGFS adalah (1:1:5:1:10). Sedangkan untuk subset data hasil seleksi fitur menggunakan RWIGFS adalah (1:1:3:1:10).

Tabel 4.28 Kinerja SVM-OW dengan RWIGFS dan MRIGFS pada data pengujian dari dataset NSL-KDD

Kinerja		**RWIGFS**		**MRIGFS**	
		OB-1	**OB-2**	**OB-1**	**OB-2**
Accuracy	Normal	98.92%	99.61%	98.92%	99.58%
	DoS	99.92%	99.91%	99.86%	99.84%
	R2L	97.00%	92.65%	97.60%	93.85%
	Probe	99.47%	98.48%	99.27%	98.03%
	U2R	77.14%	77.14%	57.14%	54.29%
	Semua	99.31%	99.55%	99.27%	99.47%
Sensitivity		99.83%	99.52%	99.73%	99.44%
Specificity		98.92%	99.61%	98.92%	99.58%
FPR		1.09%	0.39%	1.25%	0.49%
FNR		0.17%	0.49%	0.23%	0.49%
TP		40896	40783	40857	40724
TN		46673	47002	46677	46986
FP		513	184	509	200
FN		70	195	109	230

Confusion-matrix yang dihasilkan dari pelatihan dan pengujian SVM-OW menggunakan subset data hasil seleksi fitur MRIGFS dengan pendekatan OB-1 disajikan pada Tabel 4.29 - 4.30 dan *confusion-matrix* untuk pendekatan OB-2

disajikan pada Tabel 4.31 - 4.32. Sedangkan *confusion-matrix* yang dihasilkan dari pelatihan dan pengujian SVM-OW menggunakan subset data hasil seleksi fitur RWIGFS dengan pendekatan OB-1 disajikan pada Tabel 4.33-4.34 dan *confusion-matrix* untuk pendekatan OB-2 disajikan pada Tabel 4.35-4.36.

Tabel 4.29 *Confusion-matrix* dari pelatihan WSVM dengan kombinasi bobot kelas OB-1 pada 19 fitur dari MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	19929	2	166	59	1
	DoS	10	13789	2	12	0
	R2L	17	0	311	0	0
	Probe	25	2	1	3451	0
	U2R	5	0	0	0	12

Tabel 4.30 *Confusion-matrix* dari pengujian WSVM dengan kombinasi bobot kelas OB-1 pada 19 fitur dari MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	46677	7	392	97	13
	DoS	30	32069	1	14	0
	R2L	11	0	651	1	4
	Probe	53	5	2	8117	0
	U2R	15	0	0	0	20

Tabel 4.31 *Confusion-matrix* dari pelatihan WSVM dengan kombinasi bobot kelas OB-2 pada 19 fitur dari MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	20059	2	53	43	0
	DoS	20	13790	2	1	0
	R2L	27	0	301	0	0
	Probe	62	10	1	3406	0
	U2R	5	0	0	0	12

Tabel 4.32 *Confusion-matrix* dari pengujian WSVM dengan kombinasi bobot kelas OB-2 pada 19 fitur dari MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	46986	5	123	61	11
	DoS	45	32063	1	5	0
	R2L	37	0	626	0	4
	Probe	132	21	8	8016	0
	U2R	16	0	0	0	19

Tabel 4.33 *Confusion-matrix* dari pelatihan WSVM dengan kombinasi bobot kelas OB-1 pada 18 fitur dari RWIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	19939	11	138	59	10
	DoS	4	13800	1	5	3
	R2L	5	0	322	1	0
	Probe	20	0	0	3459	0
	U2R	4	0	3	1	9

Tabel 4.34 *Confusion-matrix* dari pengujian WSVM dengan kombinasi bobot kelas OB-1 pada 18 fitur dari RWIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	46673	18	343	128	24
	DoS	9	32088	0	12	5
	R2L	15	0	647	1	4
	Probe	43	0	0	8134	0
	U2R	3	1	3	1	27

Tabel 4.35 *Confusion-matrix* dari pelatihan WSVM dengan kombinasi bobot kelas OB-2 pada 18 fitur dari RWIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	20077	10	28	32	10
	DoS	13	13799	0	1	0
	R2L	24	0	301	1	2
	Probe	54	1	0	3424	0
	U2R	3	0	0	1	13

Tabel 4.36 *Confusion-matrix* dari pengujian WSVM dengan kombinasi bobot kelas OB-2 pada 18 fitur dari RWIGFS

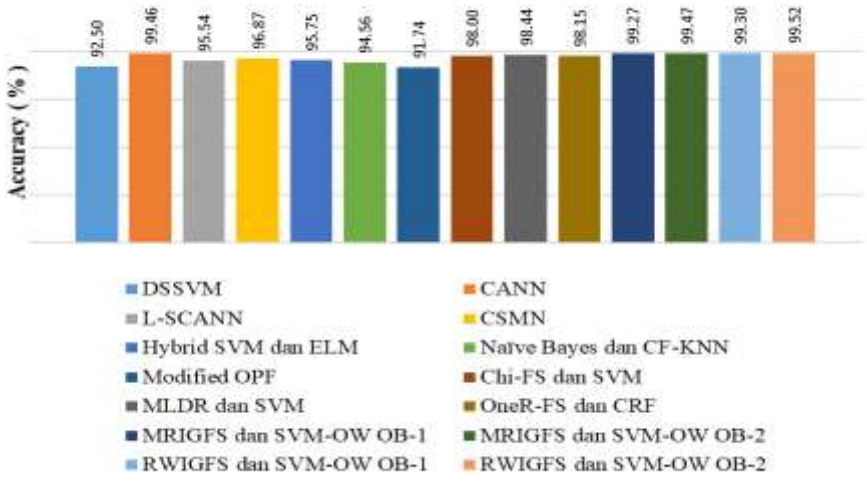
		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	47002	18	87	65	14
	DoS	23	32085	0	6	0
	R2L	44	0	618	0	5
	Probe	120	4	0	8053	0
	U2R	8	0	0	0	27

Tabel 4.37 Perbandingan kinerja dengan model IDS lain

Metode IDS	**∑ fitur**	**Accuracy (%)**					
		Overall	**Normal**	**DoS**	**R2L**	**Probe**	**U2R**
DSSVM (Guo et al., 2014)	52	92.50	98.40	97.21	6.31	87.49	3.07
CANN (Lin et al., 2015)	19	99.46(3)	97.04	99.68	57.05	87.61	3.85
L-SCANN (Ahmad & Muchammad, 2016)	19	95.54	97.50	96.90	88.35	79.43	18.18
CSMN (Muttaqien & Ahmad, 2017)	19	96.87	97.43	97.70	48.57	91.22	9.09
Hybrid SVM and ELM (Al-Yaseen et al., 2017)	---	95.75	98.13	99.54	31.39	87.22	21.93
Naïve Bayes and CF-KNN (Pajouh et al., 2017)	---	94.56	94.56	84.68	34.81	79.76	67.16(6)
Modified OPF (Bostani & Sheikhan, 2017)	---	91.74	98.55	96.89	81.13	85.92	77.98(3)
Chi-FS and multi-class SVM (Sumaiya Thaseen & Aswani Kumar, 2017)	31	98.00	99.60(1)	99.90(2)	98.70(1)	99.20(3)	73.90(5)
MLDR and multi-class SVM (Kumar et al., 2018)	---	98.44	95.74	95.99	78.66	94.97	79.77(2)
OneR-FS and CRF (Mahendiran & Appusamy, 2018)	24	98.15	98.58	98.02	96.11(4)	96.57	92.30(1)
WSVM dengan MRIGFS Kombinasi bobot (2:2:101:9:1923)	**19**	**99.27(5)**	**98.92(3)**	**99.86(3)**	**97.60(3)**	**99.27(2)**	**57.14(7)**
WSVM dengan MRIGFS Kombinasi bobot (1:1:5:1:10)	**19**	**99.47(2)**	**99.58(2)**	**99.84(4)**	**93.85(5)**	**98.03(5)**	**54.29(8)**
WSVM dengan RWIGFS Kombinasi bobot (2:2:101:9:1923)	**18**	**99.30(4)**	**98.92(3)**	**99.91(1)**	**98.17(2)**	**99.43(1)**	**52.94(9)**
WSVM dengan RWIGFS Kombinasi bobot (1:1:3:1:10)	**18**	**99.52(1)**	**99.60(1)**	**99.90(2)**	**91.77(6)**	**98.42(4)**	**76.47(4)**

Hasil uji coba diatas menunjukkan bahwa implementasi SVM-OW dengan metode normalisasi Log, kedua metode seleksi fitur, dan kedua pendekatan

optimasi bobot kelas dapat mendeteksi semua kelas serangan sehingga kondisi *completeness* terpenuhi. Secara berturutan *sensitivity*, *specificity*, *accuracy* keseluruhan, *accuracy* kelas Normal, *accuracy* kelas DoS, *accuracy* kelas R2L, *accuracy* kelas Probe, dan *accuracy* kelas U2R yang dihasilkan SVM-OW dengan metode seleksi fitur RWIGFS dan pendekatan optimasi OB-1 adalah 99.83%, 98.92%, 99.31%, 98.92%, 99.92%, 97.00%, 99.47%, dan 77.14%. Dengan urutan yang sama, kinerja yang dihasilkan ketika menggunakan pendekatan optimasi OB-2 adalah 99.52%, 99.61%, 99.55%, 99.61%, 99.91%, 92.65%, 98.48%, dan 77.14%.



Gambar 4.25 Perbandingan *accuracy* keseluruhan dengan model IDS lain

Dengan urutan yang sama, kinerja yang dihasilkan SVM-OW dengan metode seleksi fitur MRIGFS dan pendekatan optimasi OB-1 adalah 99.73%, 99.92%, 99.27%, 99.92%, 99.86%, 97.60%, 99.27%, dan 57.14%. Sedangkan untuk pendekatan optimasi OB-2 adalah 99.44%, 99.58%, 99.47%, 99.58%, 99.84%, 93.85%, 98.03%, dan 54.29%.

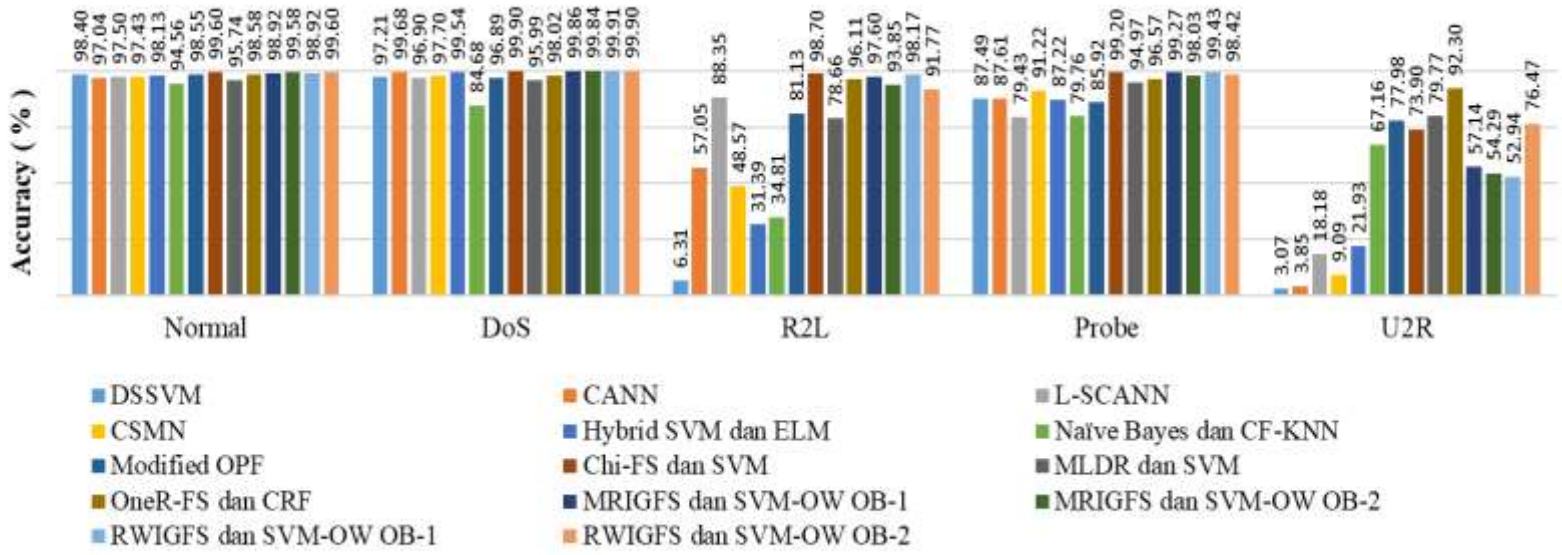
Dari hasil pengujian tersebut dapat disimpulkan bahwa metode seleksi fitur RWIGFS lebih baik dari MRIGFS pada sebagian besar ukuran kinerja yang diuji. Metode seleksi fitur RWIGFS juga menghasilkan jumlah FN (serangan yang dianggap sebagai aktivitas normal) yang lebih sedikit dibanding yang dihasilkan

oleh MRIGFS. Hal tersebut menunjukkan bahwa metode seleksi fitur RWIGFS lebih sesuai untuk diterapkan pada SVM-OW. Sedangkan analisis pada pendekatan optimasi bobot kelas yang diuji, tampak bahwa pendekatan optimasi OB-1 menghasilkan jumlah FN lebih kecil dibanding pendekatan optimasi OB-2, namun *accuracy* pada keseluruhan kelas yang dihasilkan dengan pendekatan OB-1 lebih rendah.

Selanjutnya kami akan membandingkan kinerja keempat model IDS berbasis SVM-OW tersebut dengan model IDS yang ada sebelumnya, data perbandingan kami sajikan pada Tabel 4.37, Gambar 4.25, dan Gambar 4.26. Gambar 4.25 menunjukkan perbandingan *accuracy* keseluruhan kelas dari 12 model IDS dalam bentuk grafik diagram batang. Gambar 4.26 menunjukkan perbandingan *accuracy* masing-masing kelas dari 12 model IDS dalam bentuk grafik diagram batang. Dari kiri ke kanan adalah kelas Normal, DoS, R2L, Probe, dan U2R.

Model IDS yang digunakan sebagai pembanding adalah DSSVM (Guo et al., 2014), CANN (Lin et al., 2015), L-SCANN (Ahmad & Muchammad, 2016), CSMN (Muttaqien & Ahmad, 2017), Hybrid SVM and ELM (Al-Yaseen et al., 2017), Naïve Bayes and CF-KNN (Pajouh et al., 2017), Modified OPF (Bostani & Sheikhan, 2017), Chi-FS and multi-class SVM (Sumaiya Thaseen & Aswani Kumar, 2017), MLDR and multi-class SVM (Kumar et al., 2018), dan OneR-FS and CRF (Mahendiran & Appusamy, 2018)

Tampak di Tabel 4.37 bahwa keempat model berbasis SVM-OW ini ada pada 5 urutan tertinggi dari 14 model IDS untuk kinerja *accuracy* keseluruhan kelas, *accuracy* kelas Normal, *accuracy* kelas DoS, dan *accuracy* kelas Probe. Untuk *accuracy* pada kelas R2L ada pada 6 urutan tertinggi dari 14 model IDS, sedangkan *accuracy* pada kelas U2R ada pada 9 urutan tertinggi dari 14 model IDS.



Gambar 4.26 Perbandingan *accuracy* pada masing-masing kelas dengan model IDS lain

b) Hasil uji coba dengan dataset Kyoto 2006++

Hasil optimasi kombinasi bobot kelas yang dilakukan dengan pendekatan OB-1 yang dilakukan dengan perbandingan bobot berdasarkan jumlah data pada masing-masing kelas serangan mendapatkan kombinasi (1.09:11.98). Pada Tabel 4.38 kami sajikan hasil uji coba SVM-OW dengan kombinasi bobot kelas tersebut pada data pengujian Kyoto2006++ 14 fitur. *Confusion-matrix* yang dihasilkan uji coba tersebut kami sajikan pada Tabel 4.39.

Tabel 4.38 Kinerja WSVM dengan optimasi bobot kelas OB-1 pada dataset Kyoto2006++ 14 fitur

Kinerja		**nilai**
Accuracy	Normal	99.74%
	Attack	99.95%
	Semua	99.75%
Sensitivity		99.95%
Specificity		99.74%
TP		5758
TN		63096
FP		167
FN		3

Tabel 4.39 *Confusion-matrix* dari pengujian WSVM dengan optimasi bobot kelas OB-1 pada dataset Kyoto2006++ 14 fitur

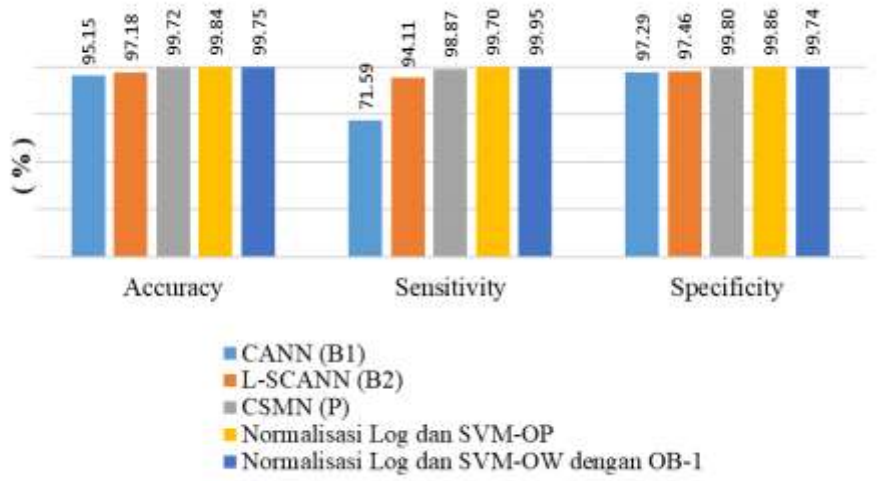
		Prediksi	
		Normal	**Attack**
Aktual	**Normal**	63096	167
	Attack	3	5758

Implementasi algoritma WSVM dengan bobot kelas menggunakan perbandingan bobot berdasarkan jumlah data pada masing-masing kelas serangan dan metode normalisasi log pada dataset 14 fitur dari Kyoto2006++ menghasilkan *accuracy*, *sensitivity*, dan *specificity* yang tinggi. Nilainya secara berturutan adalah 99.75%, 99.95%, dan 99.74%. Ketika dibandingkan dengan hasil 7 penelitian IDS yang menggunakan dataset Kyoto2006++ peringkat *accuracy, sensitivity,* dan *specificity*-nya secara berturutan adalah 1, 1, dan 2. Jika dibandingkan dengan kinerja yang dihasilkan SVM dengan optimasi kernel RBF, pendekatan ini unggul

dalam *sensitivity* namun kalah pada *accuracy* dan *specificity*. Data perbandingan kinerja kami sajikan pada Tabel 4.40.

Tabel 4.40 Perbandingan Kinerja dengan model IDS lain

Metode IDS	∑ fitur	Accuracy (%)	Sensitivity (%)	Specificity (%)
L-SCANN (Ahmad & Muchammad, 2016)	7	94.84	95.51	93.53
CANN (B1) (Muttaqien & Ahmad, 2017)	7	89.37	74.06	90.52
L-SCANN (B2) (Muttaqien & Ahmad, 2017)	7	95.32	86.03	96.01
CSMN (P) (Muttaqien & Ahmad, 2017)	7	97.47	92.59	97.84
CANN (B1) (Muttaqien & Ahmad, 2017)	14	95.15	71.59	97.29
L-SCANN (B2) (Muttaqien & Ahmad, 2017)	14	97.18	94.11	97.46
CSMN (P) (Muttaqien & Ahmad, 2017)	14	99.72	98.87	99.80(1)
Normalisasi Log dan WSVM OB-1	**14**	**99.75(1)**	**99.95(1)**	**99.74(2)**
Normalisai Log dan SVM-OP	**14**	**99.87**	**99.70**	**99.86**



Gambar 4.27 Perbandingan dengan model IDS lain untuk dataset Kyoto dengan 14 fitur

Implementasi IDS berbasis SVM-OW yang diusulkan pada dataset NSL-KDD dan dataset Kyoto2006++ dapat mencapai kondisi *completeness* karena dapat mendeteksi semua kelas serangan pada dataset yang diuji. Berikut tiga kesimpulan

yang dapat kami ambil dari uji coba model IDS berbasis SVM-OW yang kami usulkan:

1. Metode seleksi fitur RWIGFS lebih sesuai untuk dikombinasikan dengan SVM-OW. Karena pada uji coba dengan dataset NSL-KDD, penggunaan metode seleksi fitur RWIGFS pada SVM-OW menghasilkan kinerja lebih baik dibanding jika menggunakan MRIGFS pada sebagian besar ukuran kinerja yang diuji dan menghasilkan *false negative* yang lebih sedikit.
2. Penggunaan pendekatan optimasi bobot OB-1 pada SVM-OW menghasilkan jumlah *false negative* lebih sedikit dibanding pendekatan optimasi OB-2, namun *accuracy* keseluruhan kelas yang dihasilkan oleh pendekatan optimasi bobot OB-1 lebih rendah.
3. Implementasi algoritma SVM-OW dengan optimasi bobot OB-1 dan metode normalisasi log pada dataset 14 fitur dari Kyoto2006++ menghasilkan *accuracy*, *sensitivity*, dan *specificity* yang tinggi, diatas 99%. Dibandingkan dengan 7 model IDS yang ada sebelumnya yang menggunakan dataset Kyoto2006++, peringkat *accuracy*, *sensitivity*, dan *specificity*-nya secara berturutan adalah 1, 1, dan 2.

4.6. IDS berbasis Ensemble-Voting

Bagian ini menyajikan hasil eksperimen dari pengklasifikasi IV yang merupakan *ensemble-voting* dari pengklasifikasi I (L-SCANN), pengklasifikasi II (SVM-OP), dan pengklasifikasi III (SVM-OW). Uji coba dilakukan dengan menggunakan dataset NSL-KDD dan KYOTO 2006++. Untuk dataset NSL-KDD, ada dua macam *ensemble-voting* yang dilakukan. Pertama, gabungan dari L-SCANN, SVM-OP, dan WSVM dengan optimasi bobot OB-1 yang selanjutnya kami sebut dengan EV-1. Kedua, gabungan dari L-SCANN, SVM-OP, dan WSVM dengan optimasi bobot OB-2 yang selanjutnya kami sebut dengan EV-2.

Uji coba pendekatan *ensemble-voting* dilakukan menggunakan dataset pelatihan dari NSL-KDD yang terdiri dari 125973 record. Uji coba pelatihan dan pengujian dilakukan dengan metoda 10-*fold cross validation*, dimana dataset dibagi

menjadi sepuluh subset yang tidak terduplikasi, dan sembilan dari sepuluh subset digunakan untuk pelatihan dan sisanya untuk pengujian. Dengan demikian, pengklasifikasi akan dilatih dan diuji 10 kali.

a) Hasil uji coba dengan dataset NSL-KDD

Hasil uji coba pengklasifikasi IV dengan *ensemble-voting* EV-1 dan EV-2 pada dataset NSL-KDD dengan subset data hasil seleksi fitur menggunakan MRIGFS kami sajikan pada Tabel 4.41. *Confusion-matrix* hasil uji coba dengan *ensemble-voting* EV-1 disajikan pada Tabel 4.42. Sedangkan *confusion-matrix* hasil uji coba dengan *ensemble-voting* EV-2 kami sajikan pada Tabel 4.43.

Tabel 4.41 Kinerja *Ensemble-voting* EV-1 dan EV-2 pada dataset 19 fitur hasil MRIGFS

Kinerja		**EV-1**	**EV-2**	**CBC**	**SVM-OW (OB-1)**	**SVM-OW (OB-2)**	**SVM-OP**
Accuracy	Normal	99.81%	99.74%	97.05%	99.63%	98.98%	99.87%
	DoS	99.96%	99.95%	96.73%	99.93%	99.92%	99.97%
	R2L	92.96%	94.17%	65.73%	92.96%	97.79%	93.67%
	Probe	98.96%	99.53%	92.32%	98.46%	99.57%	99.67%
	U2R	63.46%	65.38%	38.46%	76.92%	80.77%	61.54%
	Semua	99.71%	99.74%	96.23%	99.57%	99.36%	99.82%
Sensitivity		99.68%	99.78%	96.21%	99.54%	99.86%	99.79%
Specificity		99.81%	99.74%	97.05%	99.63%	99.08%	99.87%
FPR		0.19%	0.26%	2.95%	0.37%	0.92%	0.13%
FNR		0.32%	0.22%	3.79%	0.46%	0.14%	0.21%
TP		58440	58503	56410	58361	58546	58508
TN		67212	67169	65356	67095	66658	67253
FP		131	174	1987	248	685	90
FN		190	127	2220	269	84	122

Tabel 4.42 *Confusion-matrix* hasil pengujian *ensemble-voting* EV-1 pada dataset 19 fitur hasil MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	67212	19	50	50	12
	DoS	12	45907	2	6	0
	R2L	66	0	925	0	4
	Probe	100	21	0	11535	0
	U2R	12	1	4	2	33

Tabel 4.43 *Confusion-matrix* hasil pengujian *ensemble-voting* EV-2 pada dataset 19 fitur hasil MRIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	67169	22	58	82	12
	DoS	13	45906	1	7	0
	R2L	52	0	937	3	3
	Probe	51	4	0	11601	0
	U2R	11	0	4	3	34

Hasil uji coba pengklasifikasi IV dengan *ensemble-voting* EV-1 dan EV-2 pada dataset NSL-KDD dengan subset data hasil seleksi fitur menggunakan RWIGFS kami sajikan pada Tabel 4.44. *Confusion-matrix* hasil uji coba dengan *ensemble-voting* EV-1 disajikan pada Tabel 4.45. Sedangkan *confusion-matrix* hasil uji coba dengan *ensemble-voting* EV-2 kami sajikan pada Tabel 4.46.

Tabel 4.44 Kinerja *Ensemble-voting* EV-1 dan EV-2 pada dataset 18 fitur hasil RWIGFS

Kinerja		**EV-1**	**EV-2**	**CBC**	**SVM-OW (OB-1)**	**SVM-OW (OB-2)**	**SVM-OP**
Accuracy	Normal	99.83%	99.77%	97.41%	99.65%	99.02%	99.85%
	DoS	99.95%	99.96%	96.27%	99.92%	99.92%	99.97%
	R2L	92.26%	94.87%	3.22%	92.76%	98.09%	94.87%
	Probe	98.87%	99.49%	93.94%	98.50%	99.54%	99.67%
	U2R	67.31%	69.23%	21.15%	76.92%	75.00%	67.31%
	Semua	99.71%	99.76%	95.90%	99.58%	99.38%	99.82%
Sensitivity		99.62%	99.78%	94.90%	99.52%	99.85%	99.82%
Specificity		99.83%	99.77%	97.41%	99.65%	99.02%	99.85%
FPR		0.17%	0.23%	2.59%	0.35%	0.98%	0.15%
FNR		0.38%	0.22%	5.10%	0.48%	0.15%	0.18%
TP		58407	58500	55642	58351	58543	58522
TN		67228	67186	65597	67110	66684	67242
FP		115	157	1746	233	659	101
FN		223	130	2988	279	87	108

Hasil uji coba diatas menunjukkan bahwa implementasi *ensemble-voting* EV-1 dan EV-2 dari tiga pengklasifikasi L-SCANN, SVM-OP, dan SVM-OW dengan metode normalisasi Log dan kedua metode seleksi fitur pada dataset NSL-KDD dapat mendeteksi semua kelas serangan sehingga kondisi *completeness* terpenuhi. Secara berturutan *sensitivity*, *specificity*, *accuracy* keseluruhan,

accuracy kelas Normal, *accuracy* kelas DoS, *accuracy* kelas R2L, *accuracy* kelas Probe, dan *accuracy* kelas U2R yang dihasilkan EV-1 dengan metode seleksi fitur RWIGFS adalah 99.62%, 99.83%, 99.71%, 99.83%, 99.95%, 92.26%, 98.87%, dan 67.31%. Dengan urutan yang sama, kinerja yang dihasilkan ketika menggunakan pendekatan EV-2 adalah 99.78%, 99.77%, 99.76%, 99.77%, 99.96%, 94.87%, 99.49%, dan 69.23%.

Tabel 4.45 *Confusion-matrix* hasil pengujian *ensemble-voting* EV-1 pada dataset 18 fitur hasil RWIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	67228	29	20	57	9
	DoS	20	45902	0	5	0
	R2L	68	0	918	3	6
	Probe	122	10	0	11524	0
	U2R	13	0	3	1	35

Tabel 4.46 *Confusion-matrix* hasil pengujian *ensemble-voting* EV-2 pada dataset 18 fitur hasil RWIGFS

		Prediksi				
		Normal	**DoS**	**R2L**	**Probe**	**U2R**
Aktual	**Normal**	67186	41	29	75	12
	DoS	15	45907	0	5	0
	R2L	44	0	944	3	4
	Probe	60	0	0	11596	0
	U2R	11	0	4	1	36

Dengan urutan yang sama, kinerja yang dihasilkan EV-1 dengan metode seleksi fitur MRIGFS adalah 99.68%, 99.81%, 99.71%, 99.81%, 99.96%, 92.96%, 98.96%, dan 63.46%. Sedangkan untuk pendekatan EV-2 adalah 99.78%, 99.74%, 99.74%, 99.74%, 99.95%, 94.17%, 99.53%, dan 65.38%.

Pada pendekatan EV-1, metode seleksi fitur MRIGFS lebih baik dari RWIGFS dalam kinerja *sensitivity*, *accuracy* kelas DoS, *accuracy* kelas R2L, *accuracy* kelas Probe, dan jumlah FN yang terjadi. Namun pada *specificity*, *accuracy* kelas Normal, dan *accuracy* kelas U2R lebih rendah kinerjanya. Sedangkan pada *accuracy* keseluruhan kelas mempunyai kinerja yang sama. Dari hasil uji coba tersebut dapat disimpulkan bahwa pada pendekatan EV-1 metode

seleksi fitur MRIGFS sedikit lebih baik dari pada RWIGFS, terutama dalam menangani FN atau serangan yang diprediksi sebagai aktivitas normal.

Pada pendekatan EV-2, metode seleksi fitur RWIGFS lebih baik dari MRIGFS dalam kinerja *specificity*, *accuracy* keseluruhan kelas, *accuracy* kelas Normal, *accuracy* kelas DoS, *accuracy* kelas R2L, dan *accuracy* kelas U2R. Namun pada *accuracy* kelas Probe dan jumlah FN memiliki kinerja yang lebih rendah. Sedangkan untuk *sensitivity* mempunyai kinerja yang sama. Dari hasil uji coba tersebut dapat disimpulkan bahwa pada pendekatan EV-2 metode seleksi fitur RWIGFS lebih baik dari pada MRIGFS.

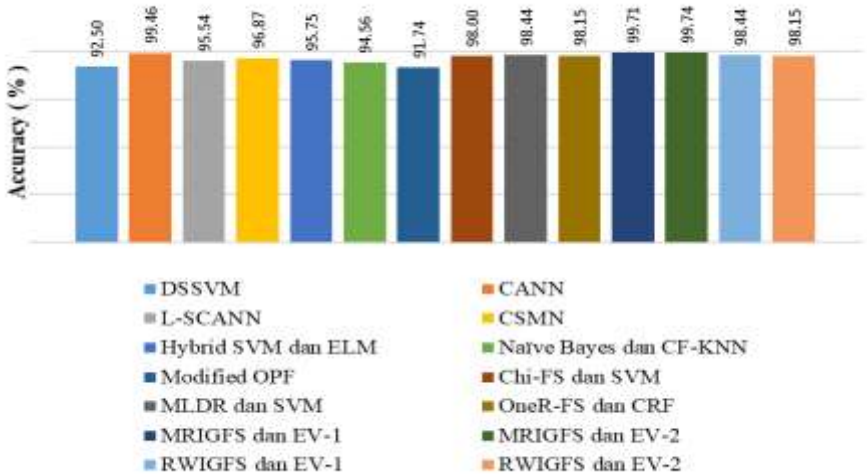
Jika dibandingkan dengan kinerja yang dihasilkan L-SCANN tanpa *ensemble-voting*, kedua pendekatan *ensemble-voting* ini unggul dalam semua ukuran kinerja yang diuji dan bisa meminimalkan jumlah FN atau serangan yang diprediksi sebagai aktivitas normal. FNR yang awalnya 5% dapat diturunkan menjadi 0.4% dan FPR yang awalnya 3% dapat diturunkan menjadi 0.3%. Hal ini menunjukkan bahwa penggabungan L-SCANN dengan SVM-OP dan SVM-OW menggunakan metode *ensemble-voting* dapat meningkatkan kinerja L-SCANN.

Selanjutnya kami akan membandingkan kinerja keempat model IDS berbasis *ensemble-voting* tersebut dengan model IDS yang ada sebelumnya, data perbandingan kami sajikan pada Tabel 4.47, Gambar 4.28, dan Gambar 4.29. Gambar 4.28 menunjukkan perbandingan *accuracy* keseluruhan kelas dari 12 model IDS dalam bentuk grafik diagram batang. Gambar 4.29 menunjukkan perbandingan *accuracy* masing-masing kelas dari 12 model IDS dalam bentuk grafik diagram batang. Dari kiri ke kanan adalah kelas Normal, DoS, R2L, Probe, dan U2R.

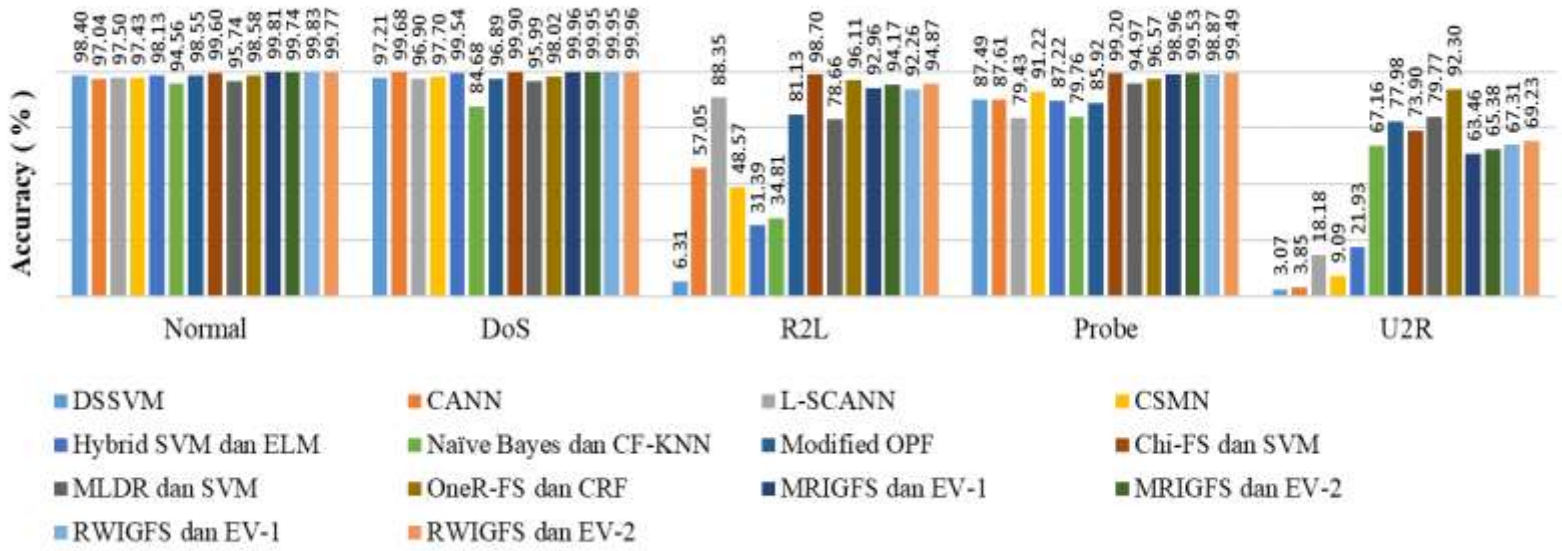
. Tampak di Tabel 4.47 bahwa keempat model berbasis *ensemble-voting* ini ada pada 4 urutan tertinggi dari 14 model IDS untuk kinerja *accuracy* keseluruhan kelas, *accuracy* kelas Normal, dan *accuracy* kelas DoS. Untuk *accuracy* kelas Probe ada pada 5 urutan tertinggi dari 14 model IDS. Untuk *accuracy* pada kelas R2L ada pada 6 urutan tertinggi, sedangkan *accuracy* pada kelas U2R ada pada 9 urutan tertinggi dari 14 model IDS.

Tabel 4.47 Perbandingan kinerja dengan model IDS lain

Metode IDS	∑ fitur	Accuracy (%)					
		Overall	Normal	DoS	R2L	Probe	U2R
DSSVM (Guo et al., 2014)	52	92.50	98.40	97.21	6.31	87.49	3.07
CANN (Lin et al., 2015)	19	99.46	97.04	99.68	57.05	87.61	3.85
L-SCANN (Ahmad & Muchammad, 2016)	19	95.54	97.50	96.90	88.35	79.43	18.18
CSMN (Muttaqien & Ahmad, 2017)	19	96.87	97.43	97.70	48.57	91.22	9.09
Hybrid SVM and ELM (Al-Yaseen et al., 2017)	---	95.75	98.13	99.54	31.39	87.22	21.93
Naïve Bayes and CF-KNN (Pajouh et al., 2017)	---	94.56	94.56	84.68	34.81	79.76	67.16(7)
Modified OPF (Bostani & Sheikhan, 2017)	---	91.74	98.55	96.89	81.13	85.92	77.98(3)
Chi-FS and multi-class SVM (Sumaiya Thaseen & Aswani Kumar, 2017)	31	98.00	99.60	99.90	98.70(1)	99.20(3)	73.90(4)
MLDR and multi-class SVM (Kumar et al., 2018)	---	98.44	95.74	95.99	78.66	94.97	79.77(2)
OneR-FS and CRF (Mahendiran & Appusamy, 2018)	24	98.15	98.58	98.02	96.11(2)	96.57	92.30(1)
Ensemble voting EV-1 dengan MRIGFS	**19**	**99.71(3)**	**99.81(2)**	**99.96(1)**	**92.96(5)**	**98.96(4)**	**63.46(9)**
Ensemble voting EV-2 dengan MRIGFS	**19**	**99.74(2)**	**99.74(4)**	**99.95(2)**	**94.17(4)**	**99.53(1)**	**65.38(8)**
Ensemble voting EV-1 dengan RWIGFS	**18**	**99.71(3)**	**99.83(1)**	**99.95(2)**	**92.26(6)**	**98.87(5)**	**67.31(6)**
Ensemble voting EV-2 dengan RWIGFS	**18**	**99.76(1)**	**99.77(3)**	**99.96(1)**	**94.87(3)**	**99.49(2)**	**69.23(5)**



Gambar 4.28 Perbandingan *accuracy* keseluruhan dengan model IDS lain



Gambar 4.29 Perbandingan *accuracy* pada masing-masing kelas dengan model IDS lain

b) Hasil uji coba dengan dataset Kyoto 2006++

Hasil uji coba pengklasifikasi IV dengan *ensemble-voting* EV-1 pada dataset Kyoto2006++ 14 fitur kami sajikan pada Tabel 4.48. *Confusion-matrix* hasil uji coba dengan *ensemble-voting* EV-1 disajikan pada Tabel 4.49.

Implementasi *ensemble-voting* dari tiga pengklasifikasi L-SCANN, SVM-OP, dan SVM-OW pada dataset 14 fitur dari Kyoto2006++ menghasilkan *accuracy*, *sensitivity*, dan *specificity* yang tinggi. Nilainya secara berturutan adalah 99.79%, 99.86%, dan 99.78%. Dibandingkan dengan hasil 7 penelitian IDS yang menggunakan dataset Kyoto2006++ peringkat *accuracy*, *sensitivity*, dan *specificity*-nya secara berturutan adalah 1, 1, dan 2. Jika dibandingkan dengan kinerja yang dihasilkan L-SCANN tanpa *ensemble-voting*, pendekatan ini unggul dalam *accuracy*, *sensitivity*, dan *specificity*-nya. Pendekatan ini juga dapat meminimalkan jumlah FN atau serangan yang diprediksi sebagai aktivitas normal. FNR yang awalnya berkisar 8% dapat diturunkan menjadi 0.14% dan FPR yang awalnya 1.36% dapat diturunkan menjadi 0.22%. Hal ini menunjukkan bahwa penggabungan L-SCANN dengan SVM-OP dan SVM-OW menggunakan metode *ensemble-voting* dapat meningkatkan kinerja L-SCANN.

Tabel 4.48 Perbandingan Kinerja untuk setiap kelas pada *ensemble-voting* untuk dataset 14 fitur Kyoto2006++

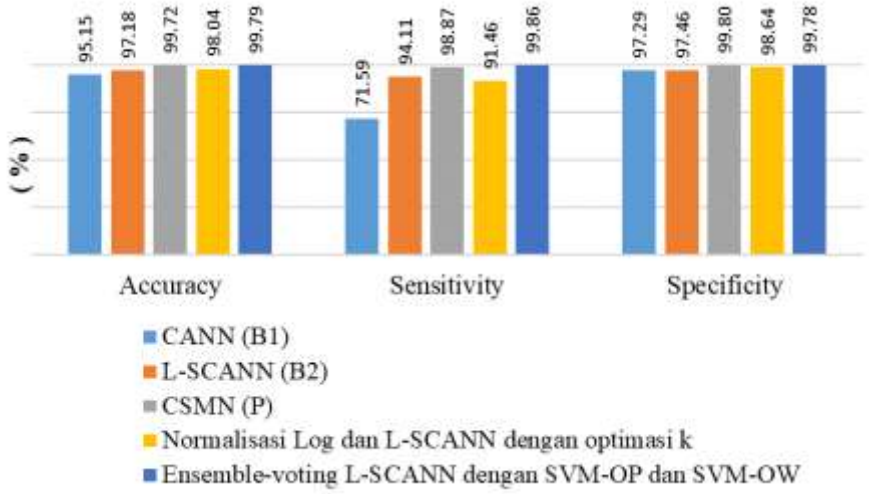
Kinerja		EV-1	CBC	SVM-OW (OB-1)	SVM-OP
Accuracy	Normal	99.78%	98.64%	99.74%	99.86%
	Attack	99.86%	91.46%	99.95%	99.70%
	Semua	99.79%	98.04%	99.75%	99.84%
Sensitivity		99.86%	91.46%	99.95%	99.70%
Specificity		99.78%	98.64%	99.74%	99.86%
FPR		0.22%	1.36%	0.26%	0.14%
FNR		0.14%	8.54%	0.05%	0.30%
TP		5753	5269	5758	5744
TN		63125	62404	63096	63172
FP		138	859	167	91
FN		8	492	3	17

Tabel 4.49 *Confusion-matrix* hasil pengujian *ensemble-voting* pada dataset 14 fitur Kyoto2006++

		Prediksi	
		Normal	**Attack**
Aktual	**Normal**	63125	138
	Attack	8	5753

Tabel 4.50 Perbandingan kinerja dengan model IDS lain

Metode IDS	**∑ fitur**	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**
L-SCANN (Ahmad & Muchammad, 2016)	7	94.84	95.51	93.53
CANN (B1) (Muttaqien & Ahmad, 2017)	7	89.37	74.06	90.52
L-SCANN (B2) (Muttaqien & Ahmad, 2017)	7	95.32	86.03	96.01
CSMN (P) (Muttaqien & Ahmad, 2017)	7	97.47	92.59	97.84
CANN (B1) (Muttaqien & Ahmad, 2017)	14	95.15	71.59	97.29
L-SCANN (B2) (Muttaqien & Ahmad, 2017)	14	97.18	94.11	97.46
CSMN (P) (Muttaqien & Ahmad, 2017)	14	99.72	98.87	99.80
Ensemble-Voting L-SCANN dengan SVM-OP dan SVM-OW	**14**	**99.79(1)**	**99.86(1)**	**99.78(2)**
Normalisasi Log dan L-SCANN dengan optimasi k	**14**	**98.04**	**91.46**	**98.64**



Gambar 4.30 Perbandingan dengan model IDS lain untuk dataset Kyoto dengan 14 fitur

Berikut kesimpulan yang dapat kami ambil dari uji coba peningkatan kinerja L-SCANN menggunakan pendekatan *ensemble-voting* dengan SVM-OP dan SVM-OW pada dataset Kyoto2006++ dan dataset NSL-KDD:

1. Penggabungan L-SCANN dengan SVM-OP dan SVM-OW menggunakan metode *ensemble-voting* dapat meningkatkan kinerja deteksi L-SCANN, baik pada *sensitivity*, *specificity*, *accuracy* keseluruhan kelas maupun *accuracy* pada masing-masing kelas. Pendekatan ini juga dapat meminimalkan jumlah *false negative* atau serangan yang diprediksi sebagai aktivitas normal.
2. Pada pendekatan EV-1 metode seleksi fitur MRIGFS menghasilkan kinerja yang sedikit lebih baik dari pada RWIGFS, terutama dalam menangani FN atau serangan yang diprediksi sebagai aktivitas normal.
3. Pada pendekatan EV-2, metode seleksi fitur RWIGFS lebih baik dari MRIGFS karena menghasilkan kinerja *specificity*, *accuracy* keseluruhan kelas, *accuracy* kelas Normal, *accuracy* kelas DoS, *accuracy* kelas R2L, dan *accuracy* kelas U2R yang lebih baik. Hanya lebih rendah pada *accuracy* kelas Probe.
4. Penggabungan L-SCANN dengan SVM-OP dan SVM-OW menggunakan metode *ensemble-voting* pada dataset Kyoto2006++ dapat meningkatkan kinerja *accuracy*, *sensitivity*, dan *specificity* dari L-SCANN. Dibandingkan dengan hasil 7 penelitian IDS yang menggunakan dataset Kyoto2006++ kinerja *accuracy*, *sensitivity*, dan *specificity* dari model ini peringkatnya secara berturutan meningkat menjadi urutan 1, 1, dan 2. Sedangkan pada dataset NSL-KDD, kinerjanya meningkat sehingga berada pada 4 urutan tertinggi dari 14 model IDS untuk kinerja *accuracy* keseluruhan kelas, *accuracy* kelas Normal, dan *accuracy* kelas DoS. Untuk *accuracy* kelas Probe, *accuracy* kelas R2L, dan *accuracy* kelas U2R secara berurutan menempati 5, 6, dan 9 urutan tertinggi dari 14 model IDS.

Halaman ini sengaja dikosongkan

BAB 5
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Hasil penelitian ini telah berhasil membangun model deteksi intrusi melalui kombinasi proses normalisasi, seleksi fitur untuk *imbalanced class*, *composite performance index*, dan pengembangan model *ensemble-voting* dari L-SCANN, SVM-OP, dan SVM-OW.

Dari hasil uji coba menggunakan dataset NSL-KDD dan Kyoto2006++, dapat disimpulkan hal-hal sebagai berikut:

1. Penggunaan metode normalisasi dan jumlah digit yang berbeda dalam pembulatan hasil normalisasi dapat merubah nilai *mutual information* dari fitur yang diproses sehingga mempengaruhi hasil seleksi fitur dan kinerja pengklasifikasi.
2. Penggunaan metode normalisasi Log dengan pembulatan hasil normalisasi diatas 3 digit desimal pada dataset NSL-KDD dan Kyoto2006++ tidak merubah nilai *mutual information* dari fitur yang diproses.
3. Kombinasi metode normalisasi Log dengan pengklasifikasi SVM, k-NN, dan L-SCANN menghasilkan kecepatan dan ketepatan klasifikasi (*accuracy*, *sensitivity*, *specificity*) yang lebih baik dibanding menggunakan metode normalisasi Min-max dan Z-score.
4. Hasil terbaik seleksi fitur pada model IDS L-SCANN adalah menggunakan MRIGFS dimana fitur terpilih dapat meningkatkan kinerja L-SCANN. Sedangkan hasil terbaik seleksi fitur pada model IDS berbasis SVM-OP dan model IDS berbasis SVM-OW didapat dengan menggunakan RWIGFS.
5. Penerapan metode normalisasi Log dan metode seleksi fitur MRIGFS pada dataset NSL-KDD serta optimasi parameter k-NN dapat meningkatkan kinerja deteksi L-SCANN dalam hal *accuracy* pada keseluruhan kelas dan *sensitivity* pada kelas minoritas (R2L dan U2R). A*ccuracy*, *sensitivity*, dan *specificity* di atas 96%, FPR dan FNR di bawah 4%, serta *sensitivity* pada *minority class* (R2L dan U2R) adalah 66% dan 38%

6. Pasangan pengklasifikasi SVM-OP dan SVM-OW dapat digunakan untuk memvalidasi hasil prediksi *false negative* atau serangan yang diprediksi sebagai aktivitas jaringan normal. Hasil uji coba dengan dataset NSL-KDD menunjukkan bahwa penerapan *ensemble-voting* menggunakan SVM-OP dan SVM-OW menurunkan nilai FNR dan FPR dari L-SCANN, FNR yang awalnya 5% dapat diturunkan menjadi 0.4% dan FPR yang awalnya 3% dapat diturunkan menjadi 0.3%. Sehingga *accuracy*, *sensitivity*, dan *specificity* semuanya meningkat menjadi diatas 99.0%, *sensitivity* pada *minority class* R2L meningkat menjadi 94%, dan *sensitivity* pada *minority class* U2R meningkat menjadi 65%. Hasil uji coba dengan dataset Kyoto2006++ juga menunjukkan pendekatan ini juga dapat meminimalkan jumlah FN atau serangan yang diprediksi sebagai aktivitas normal, FNR yang awalnya berkisar 8% dapat diturunkan menjadi 0.14% dan FPR yang awalnya 1.36% dapat diturunkan menjadi 0.22%. Sehingga *accuracy*, *sensitivity*, dan *specificity* semuanya meningkat menjadi diatas 99.0%.
7. Kombinasi metode normalisasi Log, metode seleksi fitur RWIGFS, dan pengklasifikasi SVM-OP pada dataset NSL-KDD menghasilkan *accuracy*, *sensitivity*, dan *specificity* di atas 99%, FPR dan FNR di bawah 0.25%, serta *sensitivity* pada kelas R2L dan U2R adalah 93% dan 71%.
8. Kombinasi metode normalisasi Log, metode seleksi fitur RWIGFS, dan pengklasifikasi SVM-OW pada dataset NSL-KDD menghasilkan *accuracy*, *sensitivity*, dan *specificity* di atas 99%, FPR dan FNR di bawah 0.50%, serta *sensitivity* pada kelas R2L dan U2R adalah 93% dan 71%.

5.2. Saran

Saran berkaitan pengembangan model deteksi intrusi berbasis CBC adalah sebagai berikut:

1. Diperlukan uji coba pada dataset IDS lain untuk memperkuat kesimpulan atas kinerja dari proses normalisasi, seleksi fitur untuk *imbalanced class*, dan pemodelan *ensemble-voting* dari L-SCANN, SVM-OP, dan SVM-OW.

2. Walaupun model *ensemble-voting* dari L-SCANN, SVM-OP, dan SVM-OW menghasilkan kinerja yang baik dalam hal ketepatan deteksi, namun dalam hal kecepatan masih kurang. Oleh karenanya diperlukan pengembangan lebih lanjut untuk model tersebut.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

Abdi, M. J., Hosseini, S. M., & Rezghi, M. (2012). A Novel Weighted Support Vector Machine Based on Particle Swarm Optimization for Gene Selection and Tumor Classification. *Computational and Mathematical Methods in Medicine, 2012*, 1–7. https://doi.org/10.1155/2012/320698

Accenture and Ponemon Institute. (2017). 2017 Cost of Cyber Crime Study - Insights on the Security Investments That Make a Difference, 1–56.

Ahmad, T., & Muchammad, K. (2016). L-SCANN: Logarithmic subcentroid and nearest neighbor. *Journal of Telecommunications and Information Technology*, *2016*(4), 71–80.

Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications*, *67*, 296–303. https://doi.org/10.1016/j.eswa.2016.09.041

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152). ACM.

Bostani, H., & Sheikhan, M. (2017). Modification of supervised OPF-based intrusion detection systems using unsupervised learning and social network concept. *Pattern Recognition*, *62*, 56–72. https://doi.org/10.1016/j.patcog.2016.08.027

Chen, You, Li, Y., Cheng, X., & Guo, L. (2006). Survey and Taxonomy of Feature Selection. *Inscrypt 2006,LNCS4318*, 153–167.

Chen, Yuehui, Abraham, A., & Yang, B. (2007). Hybrid flexible neural-tree-based intrusion detection systems. *International Journal of Intelligent Systems*, *22*(4), 337–352. https://doi.org/10.1002/int.20203

Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer Networks*, *31*(8), 805–822. https://doi.org/10.1016/S1389-1286(98)00017-6

Dong, Y., Liu, Y., Liang, H., Chiclana, F., & Herrera-Viedma, E. (2018). Strategic weight manipulation in multiple attribute decision making. *Omega (United Kingdom)*, *75*, 1339–1351. https://doi.org/10.1016/j.omega.2017.02.008

Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., & Witten, I. H. (2005). A Machine Learning Workbench for Data Mining, 1–10.

Galar, M., Fernández, A., Barrenechea, E., & Herrera, F. (2015). DRCW-OVO: Distance-based relative competence weighting combination for One-vs-One strategy in multi-class problems. *Pattern Recognition*, *48*(1), 28–42. https://doi.org/10.1016/j.patcog.2014.07.023

García, S. (2015). *Intelligent Systems Reference Library 72 Data Preprocessing in Data Mining*.

Guo, C., Zhou, Y., Ping, Y., Zhang, Z., Liu, G., & Yang, Y. (2014). A distance sum-based hybrid method for intrusion detection. *Applied Intelligence*, *40*(1), 178–188. https://doi.org/10.1007/s10489-013-0452-6

Hernández-Pereira, E., Suárez-Romero, J. A., Fontenla-Romero, O., & Alonso-Betanzos, A. (2009). Conversion methods for symbolic features: A comparison applied to an intrusion detection problem. *Expert Systems with Applications*, *36*(7), 10612–10617. https://doi.org/10.1016/j.eswa.2009.02.054

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, *13*(2), 415–425. https://doi.org/10.1109/72.991427

Ishizaka, A., & Siraj, S. (2018). Are multi-criteria decision-making tools useful? An experimental comparative study of three methods. *European Journal of Operational Research*, *264*(2), 462–471. https://doi.org/10.1016/j.ejor.2017.05.041

KDD Cup. (1999). The KDD Cup 1999 dataset. Retrieved April 7, 2017, from http://kdd.ics.uci.edu/databases/kddcup99/

Kumar, B. N., Raju, M. S. V. S. B., & Vardhan, B. V. (2018). Enhancing the performance of an intrusion detection system through multi- linear dimensionality reduction and Multi-class SVM. *International Journal of Intelligent Engineering and Systems*, *11*(1), 181–192. https://doi.org/10.22266/ijies2018.0228.19

Li, W., & Liu, Z. (2011). A method of SVM with normalization in intrusion detection. In *Procedia Environmental Sciences* (Vol. 11, pp. 256–262). https://doi.org/10.1016/j.proenv.2011.12.040

Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, *78*(1), 13–21. https://doi.org/10.1016/j.knosys.2015.01.009

Liu, Y., Yu, X., Xiangji, J., & An, A. (2011). Combining integrated sampling with SVM ensembles for learning from imbalanced datasets. *Information Processing and Management*, *47*(4), 617–631. https://doi.org/10.1016/j.ipm.2010.11.007

Mahendiran, A., & Appusamy, R. (2018). An intrusion detection system for network security situational awareness using conditional random fields. *International Journal of Intelligent Engineering and Systems*, *11*(3), 196–204. https://doi.org/10.22266/IJIES2018.0630.21

Minihane, N., Moreno, F., Peterson, E., Samani, R., Schmugar, C., Sommer, D., & Sun, B. (2017). McAfee Labs Threat Report: December 2017. *McAfee Labs Report*, (December), 1–13.

Muchammad, K., & Ahmad, T. (2015). Detecting Intrusion Using Recursive Clustering and Sum of Log Distance to Sub-centroid. *Procedia Computer Science*, *72*, 446–452. https://doi.org/10.1016/j.procs.2015.12.125

Muttaqien, I. Z., & Ahmad, T. (2017). Increasing performance of IDS by selecting and transforming features. In *2016 IEEE International Conference on Communication, Network, and Satellite, COMNETSAT 2016* (pp. 85–90). https://doi.org/10.1109/COMNETSAT.2016.7907422

Nardo, M., & Saisana, M. (2008). OECD/JRC Handbook on constructing composite indicators. Putting theory into practice., 1–16.

Nazer, G. M. (2011). Current Intrusion Detection Techniques in Information Technology - A Detailed Analysis. *European Journal of Scientific Research*, *65*(4), 611–624.

Ogasawara, E., Martinez, L. C., De Oliveira, D., Zimbrão, G., Pappa, G. L., & Mattoso, M. (2010). Adaptive Normalization: A novel data normalization approach for non-stationary time series. In *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2010.5596746

Pajouh, H. H., Dastghaibyfard, G. H., & Hashemi, S. (2017). Two-tier network anomaly detection model: a machine learning approach. *Journal of Intelligent Information Systems*,*48*(1) 61–74. https://doi.org/10.1007/s10844-015-0388-x

Peng, C., Wu, X., Fu, Y., & Lai, K. K. (2017). Alternative approaches to constructing composite indicators: an application to construct a Sustainable Energy Index for APEC economies. *Operational Research*, *17*(3), 747–759. https://doi.org/10.1007/s12351-016-0235-z

Pontarelli, S., Bianchi, G., & Teofili, S. (2013). Traffic-Aware Design of a High-Speed FPGA Network Intrusion Detection System. *IEEE Transactions on Computers*, *62*(11), 2322–2334. https://doi.org/10.1109/TC.2012.105

Portillo, S. P. (2014). PhD Thesis Attacks Against Intrusion Detection Networks : Evasion , Reverse Engineering and Optimal Countermeasures, (June).

Said, D., Stirling, L., Federolf, P., & Barker, K. (2011). Data preprocessing for distance-based unsupervised Intrusion Detection. In *2011 9th Annual International Conference on Privacy, Security and Trust, PST 2011* (pp. 181–188). https://doi.org/10.1109/PST.2011.5971981

Sarasamma, S. T., Zhu, Q. A., & Huff, J. (2005). Hierarchical Kohonenen Net for anomaly detection in network security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *35*(2), 302–312. https://doi.org/10.1109/TSMCB.2005.843274

Setiawan, B., Djanali, S., & Ahmad, T. (2017). A Study on Intrusion Detection Using Centroid-Based Classification. *Procedia Computer Science*, *124*, 672–681. https://doi.org/10.1016/j.procs.2017.12.204

Setiawan, B., Djanali, S., Ahmad, T., Aziz, M. N. (2019). Assessing Centroid-Based Classification Models for Intrusion Detection System Using Composite Indicators. *Procedia Computer Science*, *161*, 665–676.
https://doi.org/10.1016/j.procs.2019.11.170.

So-In, C., Mongkonchai, N., Aimtongkham, P., Wijitsopon, K., & Rujirakul, K. (2014). An evaluation of data mining classification models for network intrusion detection, 90–94. https://doi.org/10.1109/DICTAP.2014.6821663

Sommer, R., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion. *IEEE Symposium on Security and Privacy*, (May).

Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., & Nakao, K. (2011). Statistical analysis of honeypot data and building of Kyoto 2006+

dataset for NIDS evaluation. *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security - BADGERS '11*, 29–36. https://doi.org/10.1145/1978672.1978676

Sumaiya Thaseen, I., & Aswani Kumar, C. (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), 462–472.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *Proceeding of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009)*. https://doi.org/10.1109/CISDA.2009.5356528

Tsai, C.-F., Tsai, J.-H., & Chou, J.-S. (2012). Centroid-Based Nearest Neighbor Feature Representation for E-Government Intrusion Detection. In *World Telecommunications Congress (WTC), 2012* (Vol. 0091, pp. 1–6).

Tsai, C. F., & Lin, C. Y. (2010). A triangle area based nearest neighbors approach to intrusion detection. *Pattern Recognition*, *43*(1), 222–229. https://doi.org/10.1016/j.patcog.2009.05.017

Von Solms, R., & Van Niekerk, J. (2013). From information security to cyber security. *Computers and Security*, *38*, 97–102. https://doi.org/10.1016/j.cose.2013.04.004

Waikato University. (2018). Weka Data Mining Software version 3.8.3. Retrieved from https://www.cs.waikato.ac.nz/ml/weka/downloading.html

Wang, W., Zhang, X., Gombault, S., & Knapskog, S. J. (2009). Attribute normalization in network intrusion detection. *I-SPAN 2009 - The 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, 448–453. https://doi.org/10.1109/I-SPAN.2009.49

Yang, X., Song, Q., & Wang, Y. (2007). A weighted support vector machine for data classification. *International Journal of Pattern Recognition and Artificial Intelligence*, *21*(5), 961–976. https://doi.org/10.1142/S0218001407005703

Zhou, P., Ang, B. W., & Poh, K. L. (2007). A mathematical programming approach to constructing composite indicators. *Ecological Economics*, *62*(2), 291–297. https://doi.org/10.1016/j.ecolecon.2006.12.020

LAMPIRAN

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS

Nama	:	Bambang Setiawan
Tempat / Tanggal Lahir	:	Malang, 15 November 1969
Pekerjaan	:	Dosen
Pangkat / Golongan	:	Penata Tk. I / III-D
Jabatan Fungsional	:	Lektor
Masa Kerja	:	15 tahun
Alamat Kantor	:	Kampus ITS Sukolilo, 60111
Alamat Rumah	:	Perumahan YKP Pandugo I Blok Pi nomer 7 Surabaya
Institusi	:	Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember Surabaya
Email	:	setiawan@is.its.ac.id

A. Riwayat Pendidikan

- SD Sang Timur, Batu (1975-1982)
- SMP Negeri 1 Batu (1982-1985)
- SMA Negeri 8 Malang (1985-1988)
- S1 Teknik Komputer ITS (1988-1994)
- S2 Teknik Informatika ITS (1997-1999)

B. Riwayat Pekerjaan

- Pusat KUD Jawa Timur (1992-2005)
- Dosen Universitas 17 Agustus 1945 Surabaya (1995-2005)
- Dosen ITS (2005-sekarang)

C. Publikasi Ilmiah selama studi program Doktor

✓ Seminar Internasional

- Setiawan, B., Djanali, S., & Ahmad, T. (2017). *A study on intrusion detection using centroid-based classification*. In 4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia.
  https://www.sciencedirect.com/journal/procedia-computer-science/vol/124/suppl/C
- Setiawan, B., Djanali, S., Ahmad, T., & Aziz, M. N. (2019). *Assessing Centroid-Based Classification Models for Intrusion Detection System Using Composite Indicators*. In 5th Information Systems International Conference 2019, ISICO 2019, 23-24 Juli 2019, Surabaya, Indonesia.
  https://www.sciencedirect.com/journal/procedia-computer-science/vol/161/suppl/C

✓ Jurnal Internasional

- Setiawan, B., Djanali, S., & Ahmad, T. (2019). *Increasing accuracy and completeness of intrusion detection model using fusion of normalization, feature selection method and support vector machine*, International Journal Intelligent Engineering and Systems, Volume 12, Issue 4, Agustus 2019 (*published paper*). *http://www.inass.org/ContentsPapers2019-4.html*

- Setiawan, B., Djanali, S., & Ahmad, T. (2020). *Analyzing the performance of intrusion detection model using weighted one-against-one support vector machine and feature selection for Imbalanced Classess*, International Journal Intelligent Engineering and Systems (*accepted paper in January 2020*).