TESIS

HYPERPARAMETER TUNING ALGORITMA *RANDOM FOREST* UNTUK KLASIFIKASI DATA *FLOW* SERANGAN *SLOW HTTP*

HYPERPAMETER TUNING RANDOM FOREST ALGORITHM FOR DATA FLOW CLASSIFICATION SLOW HTTP ATTACK

ARDIANSYAH
D082202016

PROGRAM STUDI MAGISTER INFORMATIKA
DEPERTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2023

TESIS

HYPERPARAMETER TUNING ALGORITMA *RANDOM FOREST* UNTUK KLASIFIKASI DATA *FLOW* SERANGAN *SLOW HTTP*

Hyperpameter Tuning Random Forest Algorithm

for data Flow Classification Slow HTTP Attack

ARDIANSYAH
D082202016

PROGRAM STUDI MAGISTER INFORMATIKA

DEPERTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2023

PENGAJUAN TESIS

HYPERPARAMETER TUNING ALGORITMA *RANDOM FOREST* UNTUK KLASIFIKASI DATA *FLOW* SERANGAN *SLOW HTTP*

Tesis
Sebagai Salah Satu Syarat Untuk Mencapai Gelar S2
Program Studi S2 Informatika

Disusun Dan Diajukan Oleh

ttd

ARDIANSYAH
D082202016

Kepada

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2023

TESIS

HYPERPARAMETER TUNING ALGORITMA *RANDOM FOREST* UNTUK KLASIFIKASI DATA *FLOW* SERANGAN *SLOW HTTP*

ARDIANSYAH
D082202016

Telah dipertahankan dihadapan Panitia Ujian Tesis yang dibentuk dalam rangka penyelesaian studi pada Program Magister Teknik Informatika Fakultas Teknik Universitas Hasanuddin
Pada tanggal 22 Juni 2023
Dan dinyatakan telah memenuhi syarat kelulusan

Pembimbing Utama

Dr. Eng. Muhammad Niswar, ST., M.IT
NIP. 19730922 1999031 001

Pembimbing Pendamping

Dr. Eng. Ady Wahyudi Paundu, ST., MT
NIP. 19750313 200912 1 003

Dekan Fakultas Teknik
Universitas Hasanuddin

Prof. Dr. Eng. Ir. Muhammad Isran Ramli, ST., MT.
NIP. 19730926 200012 1 002

Ketua Program Studi
S2 Teknik Informatika

Dr.Ir. Zahir Zainuddin, M.Sc.
NIP. 19640427 198910 1 002

PERNYATAAN KEASLIAN TESIS
DAN PELIMPAHAN HAK CIPTA

Yang bertanda tangan dibawah ini

Nama : Ardiansyah

Nomor Mahasiswa : D082202016

Program Studi : S2 Teknik Informatika

Dengan ini menyatakan bahwa, tesis berjudul "*Hyperparameter Tuning* Algoritma *Random Forest* Untuk Klasifikasi Data *Flow* Serangan *Slow HTTP"* adalah benar karya saya dengan arahan dari komisi pembimbing (Dr. Eng. Muhammad Niswar, ST., M.IT dan Dr. Eng. Ady Wahyudi Paundu, ST., MT). Karya ini belum diajukan dan tidak sedang diajukan dalam bentuk apapun kepada perguruan tinggi manapun. Sumber informasi yang berasal atau kutipan dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka tesis ini. Sebagian dari tesis ini telah dipublikasikan di jurnal (*Communication and Information Technology*) sebagai Artikel dengan judul "*Slow HTTP DoS Attack Detection Using Random Forest Classifier*".

Dengan ini saya melimpahkan hak cipta dari karya tulis saya berupa tesis ini kepada Universitas Hasanuddin.

Gowa, 22 Februari 2023

Yang menyatakan

Ardiansyah

KTA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT atas rahmatnya sehingga tesis ini dapat diselesaikan.

Gagasan utama klasifikasi data *flow* serangan *Slow HTTP* dengan data flow normal untuk dapat medeteksi serangan *Slow HTTP* pada web *server* yang menjalankan service *HTTP* ataupun *HTTPS* dan membuat *dataset* yang dapat digunakan secara umum untuk mendeteksi serangan *Slow HTTP* baik pada jaringan *intranet* ataupun *internet.*

Bukan hal mudah untuk mewujudkan gagasan-gagasan tersebut dalam sebuat susunan tesis, berkat bimbingan, arahan dan motivasi berbagai pihak maka ditesis ini bisa disusun sebagaimana kaidah-kaidah yang dipersyaratkan, dan untuk itu penulis menyampaikan terima kasih kepada:

1. Dr. Eng. Muhammad Niswar, ST., M.IT sebagai Pembimbing utama, dan Dr. Eng. Ady Wahyudi Paundu, ST., MT sebagai pembimbing pendamping.
2. Dr. Amil Ahmad Ilham, ST., MIT, Dr. Eng. Zulkifli Tahir, ST., M.Sc. dan Dr. Ir. Ingrid Nurtanio, MT. sebagai komisi tim penguji
3. Rektor Universitas Hasanuddin dan Dekan Fakultas Teknik Universitas Hasanuddin yang telah menfasilitasi saya menempuh program magister serta para Dosen dan rekan-rekan dalam tim penelitian.
4. Dr. Zahir Zainuddin, M.Sc. Ketua program studi S2 Informatika
5. Kepada Pimpinan dan Staf DSITD yang telah mengizinkan kami untuk melaksanakan penelitian dilapangan, dan memberikan kesempatan untuk menggunakan fasilitas dan peralatan.

Semoga segala kebaikan dan pertolongan semuanya mendapat berkah dari Allah SWT. Dan akhirnya saya menyadari bahwa tesis ini masih jauh dari kata sempurna. Untuk itu saya dengan kerendahan hari mengharapkan saran dan kritik dari semua pihak demi mengembangan laporan penelitian ini.

Penulis
Ardiansyah

ABSTRAK

ARDIANSYAH. *Hyperparameter Tuning* Algoritma *Random Forest* Untuk Klasifikasi Data *Flow* Serangan *Slow HTTP* (dibimbing oleh **Muhammad Niswar, Ady Wahyudi Paundu**)

Serangan *Slow HTTP DoS* mengganggu layanan dengan terus membanjiri lalu lintas permintaan *HTTP* yang tidak lengkap ke *server web* yang menjalankan *HTTP* dan *HTTPS*, membuat layanan pada *web server* tidak dapat diakses. Serangan *Slow HTTP* adalah salah satu teknik dari serangan *DoS*. Pada penelitian ini mengusulkan sebuah metode untuk mengidentifikasi lalu lintas serangan *Slow HTTP Header* menggunakan *machine learning classifier*. Untuk mendeteksi serangan *DoS Slow HTTP Header*, kami menggunakan pengklasifikasi *Random Forest* untuk mengidentifikasi *traffic HTTP* normal dan *traffic DoS Slow HTTP Header*. Dalam penelitian ini, kami mengumpulkan data *traffic inbound intranet* dan *internet*, memberi label pada *dataset*, dan melakukan seleksi fitur menggunakan *Mutual Information* untuk melihat relevansi fitur terhadap *class* atau label dan *Pearson Correlation* untuk mengurangi fitur redundansi berkorelasi sangat tinggi. Hasil penelitian menunjukkan bahwa pendekatan yang kami lakukan memperoleh akurasi sebesar 100% dengan menggunakan *dataset split* 70:30. Hasil penelitian membuktikan bahwa pemilihan parameter dengan *hyperparameter Tuning* berpengaruh terhadap peningkatan akurasi *model* dari *classifier*.

Kata Kunci: *Feature Selection, Slow HTTP Attack, Denial of Service, Hyperparameter Tuning*

ABSTRACT

ARDIANSYAH. *Hyperparameter Tuning Random Forest Algorithm for Data Flow Classification Slow HTTP Attack* **(**Supervised by **Muhammad Niswar, Ady Wahyudi Paundu)**

Slow HTTP DoS Attacks interrupt the service by continuously flooding incomplete HTTP request traffic to a web server running HTTP and HTTPS, making the service inaccessible. It is one of the DoS techniques. This paper proposed a method to identify slow HTTP header attack traffic using a machine learning classifier. To detect the Slow HTTP header DoS attacks, we use a random forest classifier to identify normal HTTP and Slow HTTP header DoS traffic. In this research, we collected inbound intranet and internet traffic, labelled the dataset, and conducted feature selection using Mutual information to see the relevance of features to classes and Pearson correlation to reduce highly correlated redundancy features. The results showed that our proposed approach obtained an accuracy of 100% using data split 70:30. The research results prove that the selection of parameters with hyperparameter tuning affect to accuracy improvement.

Keywords: *Feature Selection, Slow HTTP Attack, Denial of Service, Hyperparameter Tuning*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PERMASALAHAN DAN TUJUAN PENELITIAN

I.1 Latar Belakang Masalah

Serangan *Denial of Service (DoS)* umumnya dilakukan membanjiri *server* sehingga *server* kehabisan sumber daya (*memory*, *CPU*, lalu lintas trafik), kondisi ini membuat *server* tidak dapat melayani pengguna lain (Arman, 2020). Salah satu jenis serangan yang popular digunakan untuk melumpuhkan layananan *web server* yang menjalankan *Hyper Text Transfer Protocol (HTTP)* dan cukup popular adalah serangan *Slow HTTP DoS* (Ramlan & Tarigan, 2019). Berdasarkan hasil laporan investigasi yang dilakukan oleh *Verison* pada tahun 2021 serangan *DoS* mencapai 60% dari seluruh total serangan yang ada dan *DoS* terrmasuk kedalam 20 top serangan (*Verizon Business*, 2021).

Serangan *Slow HTTP* menargetkan kelemahan dalam *application layer* tetap menggunakan koneksi yang sah ke *server* aplikasi untuk memperlambat proses layanan, serangan *Slow HTTP* dapat diklasfikasikan dalam empat kategori yaitu *Slow HTTP Header (Slowloris), Slow HTTP POST (RUDY), Slow Read,* dan *Apache Killer (Range)* (Institute of Electrical and Electronics Engineers & IEEE Communications Society, n.d.). Pada penelitian ini membahas pada jenis serangan *Slow HTTP Header (Slowloris)* jenis serangan yang mengirim permintaan *partial HTTP request* yang membuat *server* menunggu *HTTP Header* yang lengkap, penyerang terus menurus mengirim *HTTP Request header* yang tidak lengkap sebelum nilai *time out request* sebelumnya berakhir pada *web server*.

Terdapat dua jenis metode *Intrusion Detection System (IDS)* yang diusul oleh pera peneliti untuk mendeteksi serangan *Slow HTTP DoS*. Metode pertama metode *Anomaly-Base* yang medeteksi serangan berdasarkan pola unit trafik serangan. seperti dalam penelitan medeteksi serangan *Slow Read DoS* dengan menggunakan *full packet cupture* data yang kemudian diekstrak untuk mendapatkan fitur yang dapat diolah oleh *machine learning Algorithms* untuk

melihat pola dari serangan *Slow Read DoS* (Kemp et al., 2020). Hampir serupa dengan penelitan mendeteksi serangan *slow HTTP* dengan mengektrak fitur dari data *flow* trafik dengan menggunakan *tools CIC Flowmeter* untuk melihat pola berbeda antara data trafik normal dan serangan (Mon Swe et al., n.d.). Metode kedua *Signature-Based* yaitu mendeteksi serangan berdasar pola *pattern* yang dihasilkan, jenis metode ini tidak dapat mendeteksi jenis serangan diluar pola yang telah ditentukan sebelumnya. Contoh penelitan yang menggunakan metode *Signature-Base* yaitu (Arman, 2020).

Anomaly-Based intrusion detection adalah teknik yang digunakan untuk mengenali *Zero-day attacks*. Meskipun berbagai teknik deteksi *anomaly* telah dikembangkan namun ada tantangan dan masalah pada area ini, yaitu demensi data yang tinggi yang berdampak pada waktu komputasi dan salah satu pendekatan dilakukan oleh peneliti untuk menghadapi demensi data yang besar yaitu dengan memilih fitur yang optimal. Teknik seleksi fitur dengan information gain telah banyak digunakan oleh para peneliti untuk menganalisis fitur yang signifikan dan relevan yang dapat dapat meningkatkan performa pendeteksian (Kurniabudi et al., 2020a). Selain itu seleksi fitur dengan menggunakan korelasi antar atribut dapat meningkatkan nilai akurasi, *Pearson corrlelation* telah digunakan beberapa sistem deteksi untuk setiap varian *dataset* (Sugianela & Ahmad, 2020).

Ada beberapa publik *dataset* yang digunakan untuk mendeteksi *anomaly* pada jaringan komputer seperti *CICIDS-2017* digunakan pada penelitian terdahulu, didalam *dataset CICIDS-2017* semua jenis serangan *DoS* digabung menjadi satu label, setiap jenis serangan *DoS* memiliki data *flow* serangan yang berbeda hal ini akan menurunkan performa model *machine learning* untuk klasifikasi dan didalam *CICIDS-2017* kami tidak mengetahui, apakah data *flow* serangan berjalan pada *HTTP* atau *HTTPS*. kami menyiapkan *dataset* lebih spesifik pada serangan *slow HTTP Header* yang menargetkan pada *HTTP* dan *HTTPS*

Algoritma Random Forest (RF) adalah *ensamble learning* dengan teknik pembelajaran berdasarkan pohon keputusan. *RF* dibangun dari pemecahan

subset fitur dan data latih secara acak. Dan model *RF* memiliki akurasi sangat baik untuk permasalahan klasifikasi atau regresi (Kelkar & Bakal, 2020). Pada dasarnya *RF* terbentuk dari sekumpulan *Decision Tree (DT)* yang menerapkan *feature Randomness* untuk setiap *Bootstrap Aggregating* yang dihasilkan akan mengambil sejumlah fitur yang dipilih secara acak dari *data Training*. Optimasi *hyperparameter* dari pada *classifier machine learning* dapat meningkatkan performa (Moubayed et al., 2020).

Kontribusi utama dari penelitian ini adalah:

- Mengusulkan *dataset* baru yang dapat mengklasifikasi antara trafik normal dan serngan *slow HTTP Header* baik pada *HTTP* atau *HTTPS*.
- Untuk menemukan fitur terbaik dari *dataset* yang dibuat akan menggunkan teknik pemilihan fitur *Mutual Information* dan *Pearson Correlation.*
- Untuk mendapatkan akurasi terbaik menggunakan teknik *hyperparemeter tuning* pada algoritma *Random Forest*.

I.2 Rumusan Masalah

Berdasarkan latar belakang masalah, rumusan masalah adalah:

1. Bagaimana mereduksi fitur untuk meningkatkan peforma klasifikasi?
2. Bagaimana menguji *dataset* yang telah melalui tahap seleksi fitur dengan algoritma klasifikasi?
3. Bagaimana mendapatkan nilai parameter yang optimal untuk algoritma klasifikasi *Random Forest*?

I.3 Tujuan Penelitian

Berdasarkan pada rumusan masalah, tujuan penelitian adalah:

1. Menguji dan menganalisis seleksi fitur *Mutual Information* dan *Pearson Correlation* menghasilkan fitur-fitur yang optimal.

2. Menguji dan menganalisis *dataset* yang telah melalui tahap seleksi fitur dengan algoritma klasifikasi *Random Forest* untuk meningkatkan akurasi.
3. Menguji, menganalisis dengan malakukan iterasi dengan menggunakan metode *GridSearchCV* untuk mendapatkan parameter yang sesuai.

I.4 Manfaat Penelitan

Manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

1. Penelitan ini dapat memberikan pengetahuan tentang fitur-fitur terbaik dari *dataset* serangan *slow HTTP Header*.
2. Penelitian ini menghasilkan *dataset* baru yang dapat digunakan oleh penelitan berikutnya untuk menganalisi serangan *slow HTTP Header* lebih lanjut.
3. Penelitan ini dapat menjadi rujukan untuk penanganan *serangan slow HTTP Header* secara *realtime*.
4. Penelitan ini dapat memperlihatkan nilai parameter *Random Forest* yang sesuai dengan akurasi lebih baik.

BAB II
TINJAUAN PUSTAKA

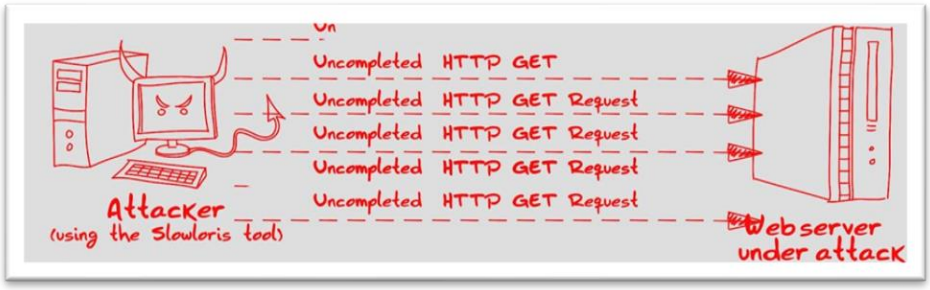
II.1 Kajian Pustaka

Kajian pustaka yang tertuang pada bab ini hasil dari studi pendahuluan yang telah dilaksanakan oleh penyusun, studi pendahuluan yang dilakukan adalah studi literatur dengan melaksanakan *review* terhadap jurnal internasional yang relevan dengan tema peneltian, mereview buku dan modul yang mendukung materi, melaksanakan *browsing* di *internet* dan juga menganalisis *video* yang relevan.

Terdapat beberapa literatur yang dijadikan sebagai kajian dalam penelitian ini, antara lain lain adalah tentang *feature selection* yaitu bagaimana melakukan reduksi fitur dan fitur yang besar untuk mendapatkan fitur-fitur yang optimal dalam penelitian ini menggunakan *Mutual Information (MI)* untuk melihat hubungan fitur terhadap kelas atau label dan menggunakan *Pearson Corrolation (PC)* untuk menghapus fitur-fitur yang berkolerasi sangat tinggi. Selain *fitur selection* dalam proses *preprocessing* data menggunakan pengskala fitur untuk membuat proses pembacaan data oleh *machine learning* lebih baik yang berdampak pada waktu komputasi nantinya. Terakhir *hyperparameter tuning* dilakukan untuk mendapat parameter terbaik pada algoritma *random forest* untuk meningkatkan akurasi dari *classifier* tersebut.

II.2 *Slow HTTP Dos Attack*

Slow HTTP Denial of Service (DoS) adalah serangan *DoS* lapisan aplikasi di mana sejumlah besar permintaan *HTTP* tidak lengkap dikirim. Ini adalah layer 7 DoS. Seranga aplikasi *DoS* adalah kelas yang baru dalam serangan *DoS* yang mengeksploitasi kelemahan dalam desain aplikasi atau implementasinya. Serangan ini lebih sulit dilacak dari pada serangan (Arman, 2020). Cara kerja *Slow HTTP* dapat dilihat pada Gambar 2.1 dibawah ini:



Gambar 2. 1 *Slow HTTP DoS Attack (youtube: Radware)*

Serangan *HTTP* lambat terutama dari tiga jenis (Arman, 2020) sebagai berikut:

1. *Slow Headers (Slowloris):* Dalam serangan *Slow Header*, seorang penyerang meluncurkan aksinya dengan bantuan alat yang disebut *Slowloris* atau sejenisnya. Alat ini membuka koneksi, kemudian mengirim *header HTTP*, menambah tetapi tidak pernah menyelesaikan permintaan. Ribuan koneksi *HTTP POST* dibuat dan mengirimkan *HTTP Header* dengan sangat lambat untuk memaksa *server web* target untuk menjaga koneksi tetap terbuka. Koneksi ini akan tetap hidup, tidak terputus dari *server* target. *Slowloris* akan mengambil semua sumber daya dari *server web* target, sehingga memblokir permintaan dari klien yang sah atau klien yang ingin mengakses *server web* tersebut.
2. *Slow Body (R-U-Dead-Yet):* Serangan *Slow Body* bekerja seperti *Slow Header*. Penyerang dengan bantuan alat yang disebut *R-U-Dead-Yet* atau sejenisnya mengirim *POST Body* yang tidak akan berakhir. Tahap serangan dimulai dengan membuat koneksi *TCP* awal ke *server web* target. Kemudian mengirimkan *header POST HTTP* terlebih dahulu seperti koneksi normal. *Header* berisi informasi ukuran tubuh paket data yang akan dikirim berikutnya. Penyerang mengirim isi pesan dengan kecepatan sangat rendah. Tetapi koneksi tetap hidup, membuat *server web* korban menunggu cukup lama. Koneksi baru dan serupa

dibuat dalam jumlah besar, menggunakan semua sumber daya *server* dan membuat koneksi yang sah menjadi tidak mungkin mengakses.

3. *Slow Read*: Dalam Serangan *Slow Read*, penyerang mengirim paket *TCP-SYN* yang valid untuk membuka koneksi dengan *server* target. Kemudian sesi yang valid dibuat di antara mereka. Selanjutnya, ia mulai meminta dokumen dari *server* target. Setelah unduhan dimulai, *host* penyerang mulai memperlambat pembacaan paket yang diterima. Kondisi ini akan berlanjut dan mengambil semua sumber daya dari *server* target. *Slow Read Attacks* selalu menggunakan paket *non-spoofed* dalam rangka untuk menahan sesi terbuka untuk jangka waktu yang lama.

II.3 *Feature Selection*

Feature selection atau seleksi fitur adalah salah satu teknik penting dan sering digunakan dalam tahap *pre-processing*. Teknik ini mengurangi jumlah fitur yang terlibat dalam menentukan suatu nilai kelas target. Fitur yang diabaikan biasanya berupa fitur yang tidak relevan dan data berlebih. Tujuan utama dari seleksi fitur ialah memilih fitur terbaik dari suatu kumpulan data fitur, Metode *Feature Selection* menjadi 3 (www.trivusi.web.id, 2022) yaitu:

II.3.1 Metode *Filter*

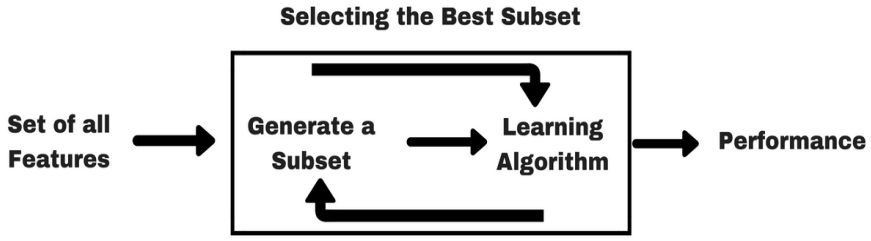
Metode *filter* mengevaluasi setiap fitur secara bebas dari *classifier* kemudian memberikan peringkat pada fitur setelah mengevaluasi dan mengambil yang unggul. Metode *filter* menerapkan ukuran statistik untuk menetapkan skor untuk setiap fitur. Fitur-fitur tersebut diberi peringkat berdasarkan skor dan dipilih untuk disimpan atau dihapus dari *dataset*. Metode ini sering bersifat univariat dan mempertimbangkan fitur secara mandiri, atau berkaitan dengan variabel dependen.

Gambar 2. 2 Metode Filter (sumber: analyticsvidhya.com)

Metode *filter* bergantung pada keunikan umum dari data yang akan dievaluasi dan memilih *subset* fitur. Metode filter menggunakan kriteria penilaian yang mencakup jarak, informasi, ketergantungan, dan konsistensi. Metode *filter* menggunakan kriteria utama teknik pemeringkatan dan menggunakan urutan peringkat untuk pemilihan variabel. Alasan untuk menggunakan metode pemeringkatan adalah kesederhanaan, menghasilkan fitur yang sangat baik dan relevan. Metode pemeringkatan akan menyaring fitur yang tidak relevan sebelum proses klasifikasi dimulai. Metode filter umumnya digunakan sebagai langkah *preprocessing* data. Penentuan dan pemilihan fitur tidak tergantung pada algoritma *Machine Learning* apa pun. Fitur memberi peringkat berdasarkan skor statistik yang cenderung menentukan korelasi fitur dengan variabel hasil. Korelasi adalah istilah yang sangat kontekstual, dan bervariasi dari satu tugas ke tugas lainnya.

II.3.2 *Metode Wrapper*

Metode *wrapper* membutuhkan satu jenis algoritma *Machine Learning* dan menggunakan kinerjanya sebagai kriteria evaluasi. Metode ini mencari fitur yang paling cocok untuk algoritma *Machine Learning* dan bertujuan untuk meningkatkan kinerja algoritma. Untuk mengevaluasi fitur, akurasi prediktif digunakan pada tugas klasifikasi.



Gambar 2. 3 *Metode Wrapper* (sumber: analyticsvidhya.com)

Metode *wrapper* didasarkan pada algoritma pencarian *greedy* karena metode ini mengevaluasi semua kemungkinan kombinasi fitur dan memilih

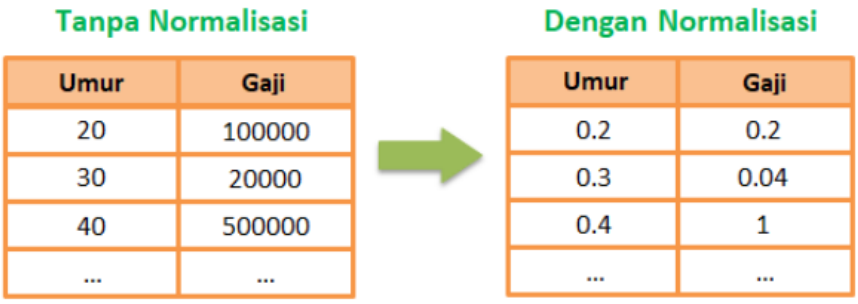
kombinasi yang menghasilkan hasil terbaik. Kelemahan dari pendekatan ini adalah pengujian semua kemungkinan kombinasi fitur dapat menjadi sangat mahal secara komputasi, terutama jika himpunan fitur sangat besar. Metode *wrapper* untuk pemilihan fitur dapat dibagi menjadi tiga kategori: Forward selection Metode seleksi berulang yang dimulai dengan fitur kosong pada model. Dalam setiap iterasi atau perulangan, kita menambahkan fitur yang memiliki pengaruh paling signifikan dalam meningkatkan model yang kita miliki. Kemudian dilanjutkan dengan penambahan variabel baru yang tidak meningkatkan kinerja model, *Backward elimination* berkebalikan dengan metode *forward selectio*n, pada metode ini model berisi semua fitur. Kemudian pada setiap iterasi atau perulangan dilakukan penghapusan fitur yang tidak meningkatkan kinerja model secara signifikan. Kita mengulangi proses ini sampai model berisi fitur yang ideal, ditandai dengan tidak ada perubahan yang ditemukan ketika dilakukan penghapusan fitur dan *Recursive Feature Elimination* metode ini adalah optimasi algoritma *greedy* yang bertujuan untuk menemukan *subset* fitur berkinerja terbaik. Pada setiap iterasi, metode ini membangun model yang dimulai dari fitur paling kiri sampai semua fitur selesai dijelajahi. Metode ini mengabaikan fitur berkinerja terbaik atau terburuk di setiap iterasi. Sebaliknya metode ini memberi peringkat fitur berdasarkan urutan eliminasinya.

Penelitian terkait seleksi fitur dengan *information gain* untuk klasifikasi dan deteksi *anomaly* (Kurniabudi et al., 2020a) pada penelitian ini akan mengeliminasi fitur dan mencari fitur yang optimal untuk memahami data, mengurangi waktu komputasi dan dimensional data yang berdampak pada peningkatan prediksi *machine learning*. hasil penelitan mendapatkan 15 fitur dari total fitur 77 dengan *feature weight information gain* diatas 0.5 dengan akurasi tertinggi yaitu 99.81% hasil menunjukkan *Random Forest* dan *J48* kemampuan baik untuk mendeteksi. Untuk penelitan seleksi fitur dengan *pearson correlation* untuk *intrusion detection system* (Sugianela & Ahmad, 2020) pada penelitian ini sama halnya dengan penelitan (Kurniabudi et al., 2020b) yaitu untuk mereduksi demensi data dan mencari optimal fitur, penelitan ini mengusulkan metode *pearson correlation* yang di

implementasikan pada aplikasi weka 3.8.3 dengan *classifier random forest* mendapatkan 18 fitur optimal yang akurasi terbaik yaitu 99.36 dengan waktu komsumsi 13.12 detik.

II.4 *Fitur Scalling Min-Max*

Normalisasi data merupakan salah satu teknik yang penting untuk dipahami dalam pra-proses data. Dalam analisis dan eksplorasi data sering kali kita menemukan banyak features atau variabel di dalam *dataset* yang akan kita analisis. Bukan hal yang jarang terjadi jika rentang nilai antara variabel tersebut sangat jauh, misalnya umur yang normalnya hanya berkisar di bawah 100 dan gaji yang kebanyakan bernilai puluhan ribu, ratusan ribu atau jutaan rupiah. (https://ilmudatapy.com/, n.d.)



Gambar 2. 4 Contoh Normalisasi Data (sumber: ilmudatapy.com)

Pada tabel di atas, kita lihat bahwa perbedaan rentang nilai dari variabel terlihat sangat jauh. Variabel umur memiliki kisaran nilai dari 0 sampai 100, sementara variabel gaji berkisar dari 0 sampai 500.000. Ini berarti variabel gaji bernilai sekitar 1000 kali lipat lebih besar dibandingkan variabel umur.

Metode normalisasi data selanjutnya adalah *Min-Max*. Cara kerjanya setiap nilai pada sebuah fitur dikurangi dengan nilai minimum fitur tersebut, kemudian dibagi dengan rentang nilai atau nilai maksimum dikurangi nilai minimum dari fitur tersebut.

$$X_{norm} = \frac{\mathrm{X} - \mathrm{X}_{min}}{\mathrm{X}_{max} - X_{min}}$$

II.5 *Hyperparameter Tuning Random Forest*

Random Forest (RF) adalah teknik pembelajaran mesin *ansambel* berdasarkan pohon keputusan. Model hutan acak dibangun dari himpunan bagian acak dari fitur dan sampel data pelatihan. Model RF memiliki akurasi yang sangat baik untuk masalah klasifikasi maupun regresi. Model ini memberikan kinerja yang baik bahkan untuk masalah dengan relevansi fitur input yang lebih sedikit. Kunci dari teknik RF adalah pemilihan sampel dan fitur secara acak untuk membuat pohon keputusan. (Kelkar & Bakal, 2020). Parameter yang pada *Random Forest* dapat dilihat pada Tabel 2.1 *parameter Random Forest.*

Tabel 2. 1 Parameter *Random Forest*

Parameters	Nilai	keterangan
n_estimators	*int, default=100*	*The number of trees in the forest.*
criterion	*{"gini", "entropy", "log_loss"}, default="gini"*	*The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see Mathematical formulation. Note: This parameter is tree-specific*

Tabel 2. 1 Parameter *Random Forest*

Parameters	Nilai	keterangan
max_depth	*int, default=None*	*The minimum number of samples required to split an internal node*
min_samples_leaf	*int or float, default=1*	*The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression*
min_weight_fraction_leaf	*float, default=0.0*	*The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node. Samples have equal weight when sample_weight is not provided*
max_features	*{"sqrt", "log2", None}, int or float, default="sqrt"*	*The number of features to consider when looking for the best split*

Tabel 2. 1 Parameter *Random Forest*

Parameters	Nilai	keterangan
max_leaf_nodes	*int, default=None*	*Grow trees with max_leaf_nodes in best-first fashion. Best nodes are defined as relative reduction in impurity. If None then unlimited number of leaf nodes.*
min_impurity_decrease	*float, default=0.0*	*A node will be split if this split induces a decrease of the impurity greater than or equal to this value*
bootstrap	*bool, default=True*	*Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree*
oob_score	*bool, default=False*	*Whether to use out-of-bag samples to estimate the generalization score. Only available if bootstrap=True.*
n_jobs	*int, default=None*	*The number of jobs to run in parallel. fit, predict, decision_path and apply are all parallelized over the trees. None means 1 unless in a*

Tabel 2. 1 Parameter *Random Forest*

Parameters	Nilai	keterangan
		joblib.parallel_backend context. -1 means using all processors. See Glossary for more details.
random_state	*int, RandomState instance or None, default=None*	*Controls both the randomness of the bootstrapping of the samples used when building trees (if bootstrap=True) and the sampling of the features to consider when looking for the best split at each node (if max_features < n_features). See Glossary for details.*
verbose	*int, default=0*	*Controls the verbosity when fitting and predicting*
warm_start	*bool, default=False*	*When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, otherwise, just fit a whole new forest. See Glossary and Fitting*

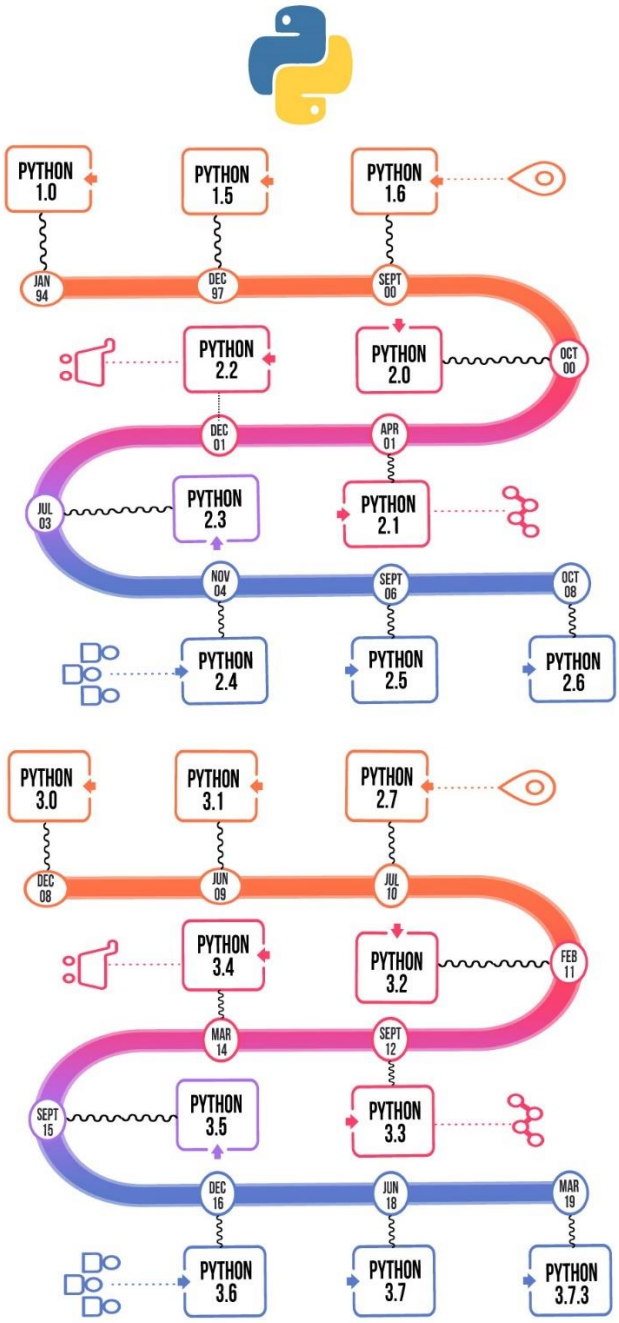
Tabel 2. 1 Parameter *Random Forest*

Parameters	Nilai	keterangan
		additional weak-learners for details.
class_weight	*{"balanced", "balanced_subsample"}, dict or list of dicts, default=None*	*Weights associated with classes in the form {class_label: weight}. If not given, all classes are supposed to have weight one. For multi-output problems, a list of dicts can be provided in the same order as the columns of y.*
ccp_alpha	*non-negative float, default=0.0*	*Complexity parameter used for Minimal Cost-Complexity Pruning. The subtree with the largest cost complexity that is smaller than ccp_alpha will be chosen. By default, no pruning is performed. See Minimal Cost-Complexity Pruning for details.*
max_samples	*int or float, default=None*	*If bootstrap is True, the number of samples to draw from X to train each base estimator.*

Dan penelitan terkait dengan *hyperparameter tuning* untuk algoritma *Random Forest* dapat dilihat pada penelitian (Kelkar & Bakal, 2020), (Moubayed et al., 2020), (Huang & Boutros, 2016), (Institute of Electrical and Electronics Engineers, n.d.) hasil pada penelitan ini menunjuk adanya pengaruh saat dilakukan tuning parameter dari random forest dimana ditemukan nilai parameter terbaik dengan akuran tertinggi. Untuk mendeteksi dan melihat pola serangan serangan *slow http* telah diteliti (Arman, 2020), (Ramlan & Tarigan, 2019) sedangkan untuk pendeteksi serangan *slow http* menggunakan *flow data* (Calvert et al., n.d.) pada penelitian ini mengumpulkan *dataset* dari data *Netflow* yang didapatkan fitur yang akan diujikan pada *classifier Random Forest, C4.5N, KNN, C4.5D, JRip, SVM, MLP* dan *Naïve Bayes* yang dihitung menggunakan rata standar deviasi *AUC* didapatkan *Random Forest* akurasi tertingi yaitu 99.89%.

II.6 *Python*

Python adalah bahasa pemrograman tujuan umum yang ditafsirkan, tingkat tinggi. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, filosofi desain *Python* menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu pemrogram menulis kode yang jelas dan logis untuk proyek skala kecil dan besar (https://id.wikipedia.org/, n.d.). Bahasa pemrograman python sudah menjadi standar untuk penelitian ilmiah eksplorasi, interaktif dan berbasisi komputasi banyaknya *library* dan dukungan yang diberikan oleh komunitas membuat Bahasa pemrograman ini sangat bayak diminati saat ini. Sehingga peningkatan perbaikan code *python* terus diperbaharui dapat dilihat dari perkembangan versinya Gambar 2.5.



Gambar 2. 5 *Python History Version (sumber: https://www.geeksforgeeks.org/history-of-python/)*

II.7 Target hasil penelitian

Target hasil dalam penelitian ini adalah penelitian ini diharapkan mampu mendapatkan *feature* yang optimal pada *dataset* yang dibuat dan *value parameter Random Forest* yang terbaik yang akan diujikan pada algoritma klasifikasi *machine learning* yang diharapkan dapat meningkatkan akurasi lebih cepat dibandingkan sebelum *dataset* melalui proses fitur seleksi.

II.8 Kerangka Pikir

Kerangka pikir dapat dilihat pada tabel 2.4, yang menjelaskan mengenai alur penelitian yang akan dilakukan.

Tabel 2. 2 Tabel Kerangka Pikir

Rumusan Masalah 1	**Rumusan Masalah 2**	**Rumusan Masalah 3**
Bagaimana mereduksi fitur untuk meningkatkan peforma klasifikasi?	Bagaimana menguji *dataset* yang telah melalui tahap seleksi fitur dengan algoritma klasifikasi?	Bagaimana mendapatkan nilai parameter yang optimal untuk algoritma klasifikasi *Random Forest*?
Tujuan Penelitian 1	**Tujuan Penelitan 2**	**Tujuan Penelitan 3**
Menguji dan menganalisis seleksi fitur *Mutual Information* dan *Pearson Correlation* menghasilkan fitur-fitur yang optimal.	Menguji dan menganalisis *dataset* yang telah melalui tahap seleksi fitur dengan algoritma klasifikasi *Random Forest* untuk meningkatkan akurasi.	Menguji, menganalisis dengan malakukan iterasi dengan menggunakan metode *GridSearchCV* untuk mendapatkan parameter yang sesuai.
Metode Penyelesaian 1	**Metode Penyelesaian 2**	**Metode Penyelesaian 3**
Untuk metode penyelesain pada	Untuk menguji apakah fitur-fitur yang telah	Menguji kemungkinan-

permasalahan seleksi fitur yaitu dengan menggunakan metode *Mutual Information gain* dengan *Pearson Correlation*	terseleksi menggunakan algoritma klasifikasi *machine learning yaitu Random Forest* untuk melihat akurasi.	kemungkin nilai parameter yang sesuai dengan melakukan iterasi menggunakna metode *GridSearchCV*
Hasil Diharapkan 1 Medapatkan fitur-fitur yang optimal untuk meningkatkan akurasi dan waktu pembuatan model *machine learning*	**Hasil Diharapkan 2** Mendapatkan *dataset* yang dapat digunakan untuk mendeteksi serangan *Slow HTTP Header Attack*	**Hasil Diharapkan 3** Mendapatkan nilai perameter terbaik dari algoritma *Random Forest* untuk meningkatkan performa klasifikasi.