

Contents list available at www.jurnal.unimed.ac.id

CESS
(Journal of Computing Engineering, System and Science)

journal homepage: <https://jurnal.unimed.ac.id/2012/index.php/cess>



Komparasi Kinerja CPU dan Memori dalam Proses Klasifikasi *Malware* Menggunakan Algoritma *Random Forest* pada Sistem Operasi Kali Linux 64-bit dan Ubuntu 64-bit

Comparison of CPU and Memory Performance in Malware Classification Process Using Random Forest Algorithm on Kali Linux 64-bit and Ubuntu 64-bit Operating Systems

Achmad Luthfan Aufer Hindami^{1*}, Dimas Rifqi Firmansyah², Christopher Ralin Anggoman³, Aqwam Rosadi Kardian⁴

^{1,2,3}Politeknik Siber dan Sandi Negara, ⁴STMIK Jakarta STI&K
Jl. Raya H. Usa, Putat Nutug, Bogor, Indonesia 16120

email: ¹achmad.luthfan@student.poltekssn.ac.id, ²dimas.rifqi@student.poltekssn.ac.id,
³christopher.ralin@student.poltekssn.ac.id, ⁴aqwam@staff.jak-stik.ac.id

ABSTRAK

Machine learning telah menjadi aspek krusial dalam keamanan siber, khususnya dalam deteksi intrusi dan klasifikasi *malware*. Namun, penerapan teknik ini memerlukan alokasi sumber daya komputasi yang signifikan. Dalam konteks ini, sistem operasi memiliki peran krusial berkaitan dengan kemampuannya dalam mengelola sumber daya komputasi. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan performa CPU dan memori dari dua sistem operasi populer, yaitu Kali Linux dan Ubuntu, dalam konteks komputasi klasifikasi *malware* menggunakan teknik dan algoritma *machine learning* untuk mengetahui sistem operasi dengan performa yang lebih baik. Keduanya diuji menggunakan model *machine learning* dan variasi dataset yang sama untuk klasifikasi *malware* menggunakan algoritma *Random Forest*. Analisis dilakukan dengan membandingkan persentase konsumsi CPU dan memori antar kedua sistem operasi. Berdasarkan hasil pengujian, ditemukan bahwa sistem operasi Kali Linux memiliki rata-rata penggunaan CPU yang lebih rendah sekitar 19,64%, dan penggunaan memori yang lebih rendah sekitar 0,06% dibandingkan dengan sistem operasi Ubuntu. Dengan demikian, dapat disimpulkan bahwa sistem operasi Kali Linux memiliki performa yang lebih baik daripada sistem operasi Ubuntu dalam hal konsumsi CPU dan memori dalam komputasi klasifikasi *malware* menggunakan teknik dan algoritma *machine learning*.

Kata Kunci: *kali linux, machine learning, malware, random forest, sistem operasi, ubuntu*

*Penulis Korespondensi:

email: achmad.luthfan@student.poltekssn.ac.id

ABSTRACT

Machine learning has become a crucial aspect of cybersecurity, particularly in intrusion detection and malware classification. However, the application of these techniques requires the allocation of significant computing resources. In this context, the operating system plays a crucial role with regard to its ability to manage computing resources. This study aims to evaluate and compare the CPU and memory performance of two popular operating systems, Kali Linux and Ubuntu, in the context of malware classification computing using machine learning techniques and algorithms to determine the operating system with better performance. Both were tested using the same machine learning model and dataset variations for malware classification using the Random Forest algorithm. The analysis was conducted by comparing the percentage of CPU and memory consumption between the two operating systems. Based on the test results, it was found that the Kali Linux operating system has an average lower CPU usage of about 19.64%, and lower memory usage of about 0.06% compared to the Ubuntu operating system. Thus, it can be concluded that Kali Linux operating system has better performance than Ubuntu operating system in terms of CPU and memory consumption in malware classification computation using machine learning techniques and algorithms.

Keywords: kali linux, machine learning, malware, random forest, operating system, ubuntu

1. PENDAHULUAN

Machine learning telah mengalami pertumbuhan dan diversifikasi yang signifikan dalam beberapa tahun terakhir. Hal ini didorong oleh banyaknya pengembangan algoritma pembelajaran baru, ledakan ketersediaan data *online*, dan komputasi berbiaya rendah [1]. *Machine learning* merupakan mesin yang didesain dan dikembangkan untuk mampu belajar dengan sendirinya tanpa arahan dari penggunanya. *Machine learning* saat ini menjadi bidang yang menarik perhatian berkaitan dengan kemampuannya untuk menghasilkan informasi dan pengetahuan dari data menggunakan suatu algoritma [2]. Bidang ini merupakan bagian dari kecerdasan buatan yang memiliki kemampuan untuk melakukan tugas seperti identifikasi data dengan tujuan prediksi [3].

Tren dalam penggunaan *machine learning* mencakup berbagai aplikasi dan domain. Hal ini mencerminkan keserbagunaan dan kemampuan beradaptasi teknik *machine learning* yang mumpuni [4]. *Machine learning* telah menjadi bagian dari keamanan siber dan deteksi *malware* yang menawarkan kemampuan canggih untuk deteksi, klasifikasi, dan pertahanan ancaman. Beberapa penelitian telah menyoroti pentingnya *machine learning* dalam meningkatkan langkah-langkah keamanan siber dan memerangi ancaman siber. Penelitian yang dilakukan oleh Zhang et al. [5] membahas pemanfaatan kecerdasan buatan secara luas, termasuk *machine learning* dan *deep learning* dalam keamanan siber, terutama dalam deteksi intrusi, deteksi *malware*, dan penyaringan *spam*. Milošević et al. [6] menekankan penggunaan teknik *machine learning* untuk klasifikasi *malware* android karena potensinya dalam membantu penyelidik forensik siber dalam mengidentifikasi dan mengkategorikan *malware*. Demikian pula Alzaylaee et al. [7] yang melakukan penelitian terkait deteksi *malware* android berbasis *deep learning* menyoroti otomatisasi dan keunggulan kecepatan yang ditawarkan oleh *machine learning* dalam proses deteksi. Dengan demikian dapat disimpulkan bahwa

machine learning dapat meningkatkan kemampuan deteksi *malware*, *breach recognition*, dan peringatan keamanan dalam organisasi [8].

Pencapaian yang didapat dari kapasitas *machine learning* yang unggul tidak dapat diperoleh secara sederhana. Hal yang perlu digarisbawahi adalah kebutuhan sumber daya komputasi dalam aplikasi *machine learning* [9]. Penerapan *machine learning* membutuhkan alokasi sumber daya komputasi yang besar. Sifat data yang berdimensi tinggi dan proses pelatihan model yang berulang-ulang pada *machine learning* membutuhkan sumber daya komputasi yang besar. Sebagai contoh, program seperti *intrusion detection system* yang berbasis *machine learning* tidak sesuai untuk perangkat dengan sumber daya yang terbatas [10]. Jumlah data yang digunakan untuk melatih model *machine learning* memiliki pengaruh yang signifikan terhadap alokasi sumber daya komputasi yang dibutuhkan. Jumlah data pelatihan yang besar dan struktur model yang kompleks dapat meningkatkan permintaan sumber daya komputasi. Metode dan teknik yang digunakan dalam suatu model *machine learning* juga membawa pengaruh terhadap permintaan serupa. Beberapa model *machine learning* tertentu seringkali membutuhkan sumber daya yang signifikan [11].

Sistem operasi dalam konteks komputasi *machine learning* memiliki peran yang krusial. Peran tersebut berkaitan dengan kemampuan sistem operasi untuk mengelola sumber daya komputasi dan mengoptimalkan kinerja sistem. Sistem yang menerapkan *machine learning* seringkali membutuhkan skalabilitas dan fleksibilitas yang dapat beradaptasi dengan pemrosesan paralel dan terdistribusi [12]. Sistem tersebut harus mampu mengatasi lingkungan dengan data berskala besar yang membutuhkan alokasi dan pengelolaan sumber daya yang efisien [13]. Sistem operasi juga memainkan peranan penting dalam pengembangan algoritma *machine learning* berskala besar [14]. Hal ini penting untuk memastikan bahwa model *machine learning* dilatih secara efektif [13]. Sistem operasi mampu menyediakan anggaran ruang dan waktu yang efisien untuk sistem *machine learning*. Hal ini memungkinkan sistem *machine learning* untuk beroperasi sesuai dengan harapan dan persyaratan pemangku kepentingan [1]. Dengan demikian, sistem operasi sangat penting untuk mendukung eksekusi algoritma *machine learning* dan mengoptimalkan kinerjanya secara nyata [15].

Kinerja sistem operasi dalam komputasi *machine learning* dipengaruhi oleh berbagai sumber daya yang dikelola. Sumber daya tersebut termasuk operasi CPU yang intensif dan bandwidth memori yang merupakan hal krusial dalam rangka mencapai tingkatan kinerja yang diharapkan [16]. Penelitian sebelumnya telah membahas seberapa baik *machine learning* dapat digunakan dalam sistem deteksi intrusi berbasis anomali. Hasilnya menitikberatkan pada berbagai faktor yang mempengaruhi seperti waktu yang dibutuhkan untuk pelatihan dan pengujian, waktu deteksi, serta penggunaan CPU dan memori [17]. Hasil identifikasi terkait tantangan dalam optimalisasi *machine learning* juga menyoroti peran krusial sumber daya komputasi, termasuk CPU dan memori [18]. Berdasarkan analisis tersebut dibutuhkan sistem operasi yang mampu mengalokasikan dan mengelola CPU dan memori secara efisien agar komputasi *machine learning* berjalan optimal.

Terdapat banyak jenis sistem operasi yang beredar dan digunakan saat ini. Masing-masing sistem operasi dikembangkan dengan spesialisasi tersendiri dan menawarkan performa yang berbeda. Beberapa sistem operasi memberikan kinerja CPU yang baik, seperti Kali Linux dan Ubuntu. Kali Linux adalah distribusi Linux turunan Debian yang dirancang untuk *digital forensic* dan *penetration testing* [19]. Dari segi performa, Kali Linux telah mendapat perhatian khusus karena efisiensinya dalam menjalankan pengujian penetrasi keamanan [19].

Pernyataan tersebut mengindikasikan kemampuan yang baik oleh Kali Linux dalam alokasi sumber daya komputasinya [20]. Sedangkan Ubuntu adalah sistem operasi yang banyak digunakan dan dikenal karena keserbagunaan dan kinerjanya. Ubuntu dapat diinstal pada berbagai sistem komputer, termasuk *virtual machine* [21]. Performa Ubuntu telah dievaluasi dan menunjukkan kesesuaiannya untuk pengujian dan analisis empiris [22]. Selain itu, Ubuntu telah digunakan dalam aplikasi *machine learning*, termasuk melakukan *porting* perangkat lunak untuk *machine learning* ke sistem operasi yang berbeda [23].

Pada penelitian ini, kinerja sistem operasi diuji dengan melakukan komputasi *machine learning* berupa klasifikasi *malware*. Tiap sistem operasi akan menjalankan model *machine learning* yang sama yaitu model klasifikasi dengan dataset tentang *malware*. Hasil pengujian kemudian akan dibandingkan dalam bentuk persentase konsumsi CPU dan konsumsi memori antar sistem operasi. Penelitian ini bertujuan untuk mengetahui sistem operasi dengan performa CPU dan memori yang lebih baik dalam komputasi klasifikasi *malware* menggunakan teknik dan algoritma *machine learning*.

2. DASAR/TINJAUAN TEORI

2.1. *Supervised learning*

Supervised learning adalah konsep fundamental dalam *machine learning* dimana algoritma belajar dari data pelatihan berlabel untuk membuat prediksi atau keputusan. Model dilatih menggunakan dataset yang berisi contoh-contoh yang sudah diberi label beserta jawaban yang benar. Prosesnya melibatkan pelatihan model pada pasangan input-output. Konsep dasarnya adalah membuat model dapat memetakan input ke output, sehingga model dapat membuat prediksi yang akurat pada data baru yang belum pernah dilihat sebelumnya. Model ini kemudian diuji pada sekumpulan data berlabel yang terpisah untuk mengevaluasi kinerjanya. Oleh karena itu, keberhasilan *supervised learning* sangat bergantung pada ketersediaan data pelatihan yang terannotasi dengan baik [24].

Supervised learning mencakup berbagai algoritma dan teknik, dan telah diterapkan secara luas di berbagai domain. Beberapa contoh penerapan *supervised learning* antara lain klasifikasi, regresi, deteksi anomali, dan sebagainya. Dalam penelitian ini jenis *supervised learning* model klasifikasi digunakan sebagai model *machine learning* yang dijalankan pada sistem operasi Kali Linux dan Ubuntu.

2.2. Klasifikasi

Klasifikasi merupakan salah satu model dalam *machine learning* jenis *supervised learning*. Klasifikasi menempatkan suatu objek atau data ke dalam salah satu kelas atau kategori berdasarkan fitur-fitur tertentu. Model ini banyak digunakan dalam berbagai domain baik itu pemrosesan data komputer atau bahkan teknologi medis. Sesuai dengan jenisnya yang termasuk *supervised learning*, proses klasifikasi biasanya melibatkan pelatihan model pada data berlabel untuk membuat prediksi pada data baru yang tidak berlabel. Model ini terus mengalami perkembangan dengan adanya eksplorasi klasifikasi *semi-supervised*. Hal ini menunjukkan adanya upaya penelitian yang sedang berlangsung untuk meningkatkan kinerja klasifikasi [25]. Pada penelitian ini digunakan model klasifikasi *malware* untuk mengkategorikan suatu *file* sebagai *malware* atau bukan *malware*.

2.3. Algoritma *Random Forest*

Random forest adalah salah satu algoritma yang kuat dan banyak digunakan dalam *machine learning*. Algoritma ini termasuk jenis metode pembelajaran ensemble yang menggabungkan beberapa pohon keputusan untuk membuat prediksi. Setiap pohon di dalam *random forest* bergantung pada nilai vektor acak yang disampel secara independen dan dengan distribusi yang sama untuk semua pohon yang terbentuk. Keacakan dan keragaman di antara pohon-pohon ini membantu mengurangi overfitting dan meningkatkan kinerja model secara keseluruhan. *Random forest* telah berhasil diterapkan di berbagai bidang, termasuk ekologi, biologi, ilmu komputer, dan kedokteran [26] [27] [28].

Salah satu keunggulan utama dari algoritma *Random Forest* adalah kemampuannya untuk menangani data dalam jumlah besar dengan dimensi yang tinggi, sehingga cocok untuk tugas klasifikasi yang kompleks [29]. Hal tersebut berkontribusi pada penggunaan algoritma ini secara luas dalam aplikasi praktis. Dalam penelitian ini, algoritma *Random Forest* digunakan untuk melatih model klasifikasi *malware* pada dataset yang berisi ribuan baris data *file* yang telah disediakan peneliti.

2.4. *Tools* htop

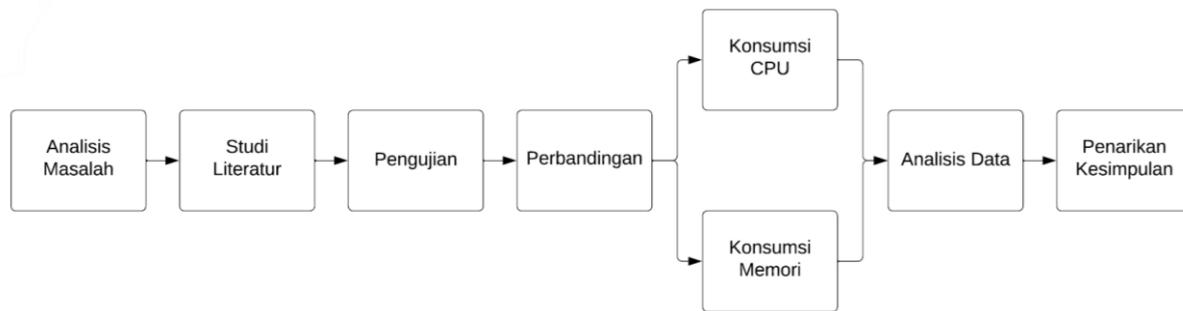
Htop merupakan aplikasi pemantauan proses yang bersifat realtime dan interaktif untuk sistem operasi Linux/Unix. . Htop digunakan bersama utilitas Linux dasar lainnya seperti Top, lostat, dan Perf, serta BCC (BPF Compiler Collection) untuk penelusuran kernel dan pelacakan status eksekusi proses di dalam sistem operasi [30]. Sebenarnya Htop merupakan alternatif dari perintah Top. Perintah Top sendiri adalah alat default pemantauan proses yang telah terinstal secara pre-instal pada seluruh sistem operasi Linux. Htop memiliki banyak fitur yang ramah pengguna dan sangat membantu untuk melakukan pemantauan proses yang berjalan serta beban sumber daya komputasi termasuk CPU dan memori. Dalam penelitian ini, Htop akan digunakan untuk melakukan pemantauan terhadap penggunaan CPU dan memori pada sistem operasi Kali Linux dan Ubuntu ketika menjalankan model klasifikasi *malware*.

3. METODE

Metode penelitian menjelaskan segala teknik dan pendekatan yang digunakan dalam pelaksanaan penelitian [31]. Setiap penelitian menggunakan metode penelitian yang berbeda menyesuaikan dengan jenis penelitian yang digunakan. Secara umum, metode penelitian dapat diklasifikasikan ke dalam dua kategori utama, yakni metode deskriptif dan metode analisis. Pada metode deskriptif peneliti tidak memiliki kendali langsung terhadap variable penelitian sehingga peneliti hanya dapat memberikan laporan tentang keadaan yang sedang berlangsung atau yang sudah terjadi dalam penelitian [31]. Sedangkan pada metode analisis peneliti menggunakan fakta atau informasi yang telah didapatkan untuk melakukan analisis dan merumuskan kesimpulan atas hal tersebut [31].

3.1. Tahapan Penelitian

Penelitian ini menerapkan perbandingan kuantitatif terhadap persentase konsumsi CPU dan konsumsi memori pada dua sistem operasi setelah menjalankan komputasi *machine learning* dengan menerapkan model clustering pada kedua sistem operasi tersebut. Tahapan metode penelitian yang digunakan dalam penelitian ini mencakup analisis masalah, studi literatur, pengujian, perbandingan, dan penarikan kesimpulan.



Gambar 1. Diagram alir tahapan penelitian

3.2. Analisis Masalah

Analisis masalah dilakukan untuk memahami permasalahan yang diangkat, kemudian diolah dengan tujuan menetapkan metode analisis dan menguji alternatif solusi yang diajukan untuk mengatasi masalah tersebut. Permasalahan yang diangkat dalam penelitian ini adalah masalah pada CPU yang mengakibatkan penurunan performa komputer dalam melakukan komputasi. Kinerja CPU dipengaruhi oleh sistem operasi yang dijalankan yang mana akan mengatur dan mengalokasikan penggunaan sumber daya komputer termasuk CPU. Dalam hal internal CPU terdapat banyak faktor yang dapat mempengaruhi CPU, termasuk *instruction set*, *clock speed*, *bandwidth*, kecepatan *Front Side Bus* (FSB), dan memori.

Penelitian ini berfokus pada permasalahan bandwidth data dan memori serta clock speed suatu komputer saat menjalankan komputasi yang membutuhkan sumber daya besar seperti model clustering. Hal ini dapat terjadi karena banyaknya data yang diolah komputer dan operasi-operasi pemrosesan yang dilakukan secara paralel. Kuantitas variabel tersebut akan membuat clock speed dan FSB menurun, memori menjadi penuh, dan berakhir dengan penurunan performa komputer. Studi kasus pada penelitian ini adalah ketika konsumsi CPU dan/atau memori menyentuh titik yang tinggi maka berpotensi mengakibatkan komputer menjadi macet atau berhenti bekerja.

3.3. Studi Literatur

Studi literatur bertujuan untuk membekali penelitian dengan dasar teoritis yang kuat dan memungkinkan peneliti untuk lebih memahami faktor-faktor yang mempengaruhi penelitian tersebut. Proses studi literatur dilakukan secara cermat melalui pencarian, pembacaan, dan pemahaman mendalam terhadap teori yang relevan. Proses tersebut dilakukan dengan acuan dari berbagai sumber yang tersedia.

Sumber yang menjadi bahan studi adalah sumber ilmiah yang valid dan terakreditasi. Peneliti menekankan penggunaan sumber-sumber ilmiah yang telah diakui dan diakreditasi oleh komunitas ilmiah. Pemilihan sumber-sumber ini didasarkan pada reputasi jurnal, reputasi penulis, dan metode penelitian yang diaplikasikan. Hal ini bertujuan untuk memastikan kredibilitas penelitian.

Sumber-sumber ilmiah yang dipilih mencakup artikel dari jurnal ter indeks, konferensi ilmiah, dan buku-buku yang telah melalui proses *review*. Keberlanjutan dan relevansi informasi dari sumber-sumber ini dianggap sebagai aspek penting dalam menyusun dasar literatur yang akurat dan terpercaya. Hal ini bertujuan untuk meminimalkan risiko adanya bias atau distorsi informasi yang dapat muncul dari sumber-sumber yang tidak ter verifikasi.

3.4. Pengujian

CPU adalah pusat komputasi dari komputer yang memiliki fungsi utama untuk memproses data. Permasalahan terjadi ketika konsumsi CPU terlalu tinggi sehingga komputer akan berhenti bekerja. Pada penelitian ini dilakukan pengujian secara kuantitatif dengan menjalankan model clustering dalam jumlah dataset tertentu.

Model klasifikasi *malware* yang akan dijalankan pada masing-masing sistem operasi Kali Linux dan Ubuntu ditulis dalam bahasa pemrograman python sehingga memiliki ekstensi .py. Sedangkan dataset yang digunakan untuk melatih model klasifikasi *malware* merupakan dataset berekstensi .csv yang diperoleh dari Kaggle sehingga tidak akan bermasalah dengan kebijakan privasi pihak manapun. Dataset tersebut berisi ribuan baris data *file* yang merupakan *malware* dan bukan *malware* dengan berbagai macam fitur nya. Ketika model klasifikasi dijalankan, peneliti melakukan pemantauan terhadap persentase konsumsi CPU dan memori hingga proses klasifikasi selesai dan menghasilkan *output*.

3.5. Perbandingan

Perbandingan diawali dengan mengambil data dari pengujian yang telah dilakukan. Perbandingan dilakukan dengan membandingkan data konsumsi penggunaan CPU dan penggunaan memori yang diperoleh selama pengujian sistem operasi Kali Linux dan Ubuntu Tujuan dari perbandingan ini adalah untuk menilai sistem operasi yang menunjukkan performa CPU yang unggul.

3.6. Analisis Data

Analisis data merupakan tahapan penting dalam penelitian dimana data konsumsi CPU dan penggunaan memori yang telah dibandingkan disusun dan dieksplorasi untuk memberikan deskripsi yang lebih mendalam. Proses ini bertujuan untuk mendapatkan pola-pola yang signifikan, hubungan antar variabel, serta perbedaan kinerja CPU pada kedua sistem operasi yang menjadi fokus penelitian. Teknik statistik seperti perhitungan rata-rata akan digunakan dalam tahapan ini untuk mendapatkan gambaran yang lebih komprehensif. Selain itu, analisis data melibatkan visualisasi data seperti grafik dan tabel untuk memberikan pemahaman yang lebih intuitif.

3.7. Penarikan Kesimpulan

Kesimpulan diperoleh setelah analisis data hasil pengujian dan perbandingan performa CPU dan penggunaan memori dilakukan. Penarikan kesimpulan mengacu pada data kuantitatif yang direpresentasikan dalam bentuk grafik dan tabel. Dengan demikian, kesimpulan ini menjadi dasar untuk pertimbangan dalam menentukan sistem operasi yang lebih efisien dalam mengelola sumber daya seperti CPU dan memori dalam konteks komputasi klasifikasi *malware*.

4. HASIL DAN PEMBAHASAN

Bagian hasil dan pembahasan dalam penelitian ini menjadi landasan untuk menguraikan temuan-temuan yang diperoleh selama proses pengujian sistem operasi Kali Linux dan Ubuntu, khususnya dalam konteks performa CPU dan memori ketika menjalankan komputasi klasifikasi *malware* menggunakan algoritma *random forest*. Pada bagian ini akan disajikan interpretasi dalam bentuk grafik dan tabel dari hasil-hasil tersebut. Selanjutnya, temuan yang diperoleh

akan dianalisis untuk menentukan sistem operasi yang menunjukkan kinerja lebih unggul baik dari segi penggunaan CPU maupun penggunaan memori.

4.1 Pengumpulan Data

Pengumpulan data dilakukan dengan melakukan menjalankan model klasifikasi *malware* pada sistem operasi Kali Linux dan Ubuntu. Dataset yang digunakan dalam model klasifikasi tersebut dibuat bervariasi dengan jumlah 500 dataset, 1000 dataset, 5000 dataset, 10000 dataset, dan 15000 dataset. Sumber daya komputasi yang digunakan akan dipantau menggunakan htop. Setiap pengujian dengan salah satu variasi dataset selesai, sistem operasi dimatikan. Hal ini dilakukan untuk memastikan kondisi sistem operasi tetap sama selama pengujian dan menjaga validitas hasil data yang diperoleh.

4.1.1 Data Performa Sistem Operasi Kali Linux

Tabel 1. Data Performa Sistem Operasi Kali Linux

Sistem Operasi	Jumlah Dataset	Persentase Konsumsi CPU	Persentase Konsumsi Memori
Kali Linux	500	85,10%	2,50%
	1000	114,00%	2,70%
	5000	114,10%	4,50%
	10000	113,30%	6,20%
	15000	113,20%	8,20%

Tabel 1 merupakan data hasil pengujian proses klasifikasi *malware* menggunakan algoritma *random forest* pada sistem operasi Kali Linux. Data pada tabel 1 menunjukkan adanya kenaikan yang signifikan terhadap konsumsi CPU ketika menggunakan variasi 1000 dataset, yaitu kenaikan sebesar 28,90% dari variasi 500 dataset. Kemudian pada variasi dataset berikutnya tidak terjadi kenaikan yang serupa. Pada variasi dataset berikutnya terdapat kenaikan dan penurunan maksimal 1% pada masing-masing pengujian. Sedangkan konsumsi memori hampir secara merata mengalami kenaikan sekitar 2% dari variasi sebelumnya, kecuali variasi 1000 dataset yang hanya mengalami kenaikan sebesar 0,20% dari variasi 500 dataset.

4.1.2 Data Performa Sistem Operasi Ubuntu

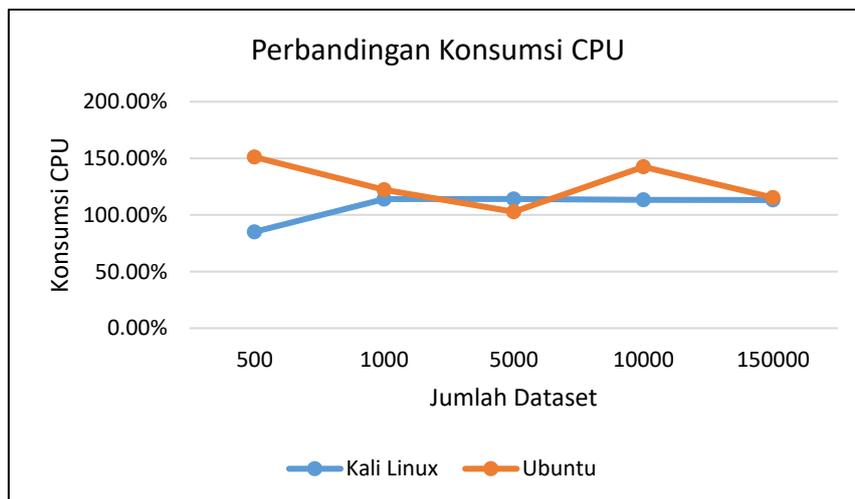
Tabel 2. Data Performa Sistem Operasi Ubuntu

Sistem Operasi	Jumlah Dataset	Persentase Konsumsi CPU	Persentase Konsumsi Memori
Ubuntu	500	151,10%	2,50%
	1000	122,20%	2,70%
	5000	102,80%	3,90%
	10000	142,50%	6,60%
	15000	115,30%	8,70%

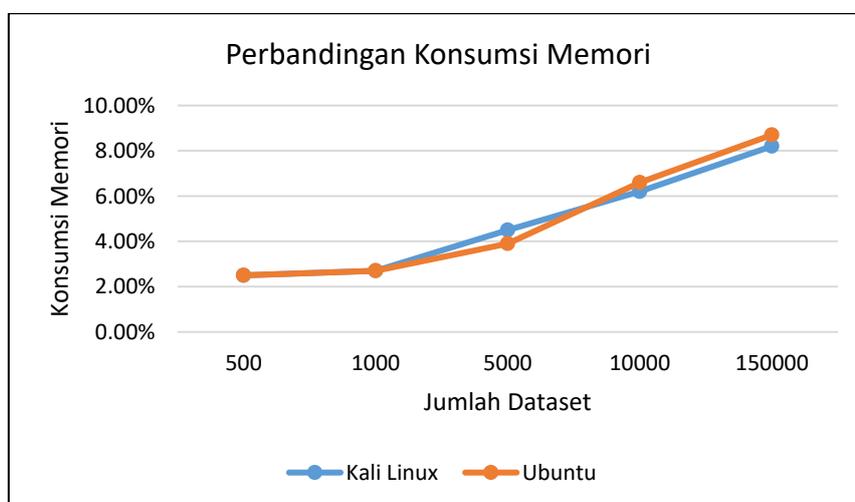
Tabel 2 merupakan data hasil pengujian proses klasifikasi *malware* menggunakan algoritma *random forest* pada sistem operasi Ubuntu. Data pada tabel 2 menunjukkan adanya penurunan yang signifikan terhadap konsumsi CPU ketika menggunakan variasi 1000 dataset, yaitu penurunan sebesar 28,90% dari variasi 500 dataset. Kemudian pada variasi dataset berikutnya juga terjadi kenaikan dan penurunan yang serupa. Namun, kenaikan yang signifikan terjadi pada variasi 10000 dataset yaitu sebesar 39,50%. Sedangkan konsumsi memori mengalami kenaikan yang cukup signifikan pada variasi 10000 dataset yaitu sebesar 2,70% dari variasi dataset sebelumnya.

4.2 Analisis Data

Analisis data menjadi tahap yang dilakukan untuk sampai pada penarikan kesimpulan. Analisis data dilakukan untuk mengetahui hasil performa sistem operasi Kali Linux dan Ubuntu dalam melakukan komputasi klasifikasi *malware* menggunakan algoritma *random forest*. Kemudian dilakukan perbandingan terhadap hasil analisis data tersebut untuk menarik kesimpulan dari penelitian.



Gambar 2. Grafik Perbandingan Konsumsi CPU



Gambar 3. Grafik Perbandingan Konsumsi Memori

Gambar 2 dan 3 menampilkan visualisasi data yang dikumpulkan dari tabel 1 dan 2. Grafik pada kedua gambar ini menunjukkan perbandingan konsumsi CPU dan memori pada pengujian terhadap sistem operasi Kali Linux dan Ubuntu. Berdasarkan grafik tersebut dapat diketahui adanya perbedaan konsumsi CPU dan memori pada masing-masing sistem operasi. Grafik pada gambar 2 memberi gambaran bahwa sistem operasi Ubuntu dominan dalam aspek konsumsi CPU. Hal ini menunjukkan bahwa mean (rata-rata) konsumsi CPU sistem operasi Ubuntu lebih tinggi daripada sistem operasi Kali Linux. Sedangkan grafik pada gambar 3 menggambarkan tidak terlalu terlihat salah satu sistem operasi yang lebih dominan. Hal ini menunjukkan bahwa mean konsumsi memori pada kedua sistem operasi relatif tidak berbeda jauh. Berdasarkan analisis tersebut dilakukan perhitungan mean untuk mendapatkan nilai eksak dari data performa yang diperoleh.

Tabel 3. Mean Performa Sistem Operasi Kali Linux dan Ubuntu

Sistem Operasi	Persentase Konsumsi CPU	Persentase Konsumsi Memori
Kali Linux	107,14%	126,78%
Ubuntu	4,82%	4,88%

Tabel 3 merupakan hasil perhitungan mean dari data performa sistem operasi Kali Linux dan Ubuntu setelah pengujian. Perhitungan dilakukan terhadap konsumsi CPU dan memori sehingga diperoleh mean kedua aspek tersebut pada masing-masing sistem operasi. Hasil perhitungan menunjukkan bahwa konsumsi CPU sistem operasi Kali Linux lebih rendah 19,64% daripada konsumsi CPU sistem operasi Ubuntu. Sedangkan konsumsi memori sistem operasi Kali Linux lebih rendah 0,06% daripada konsumsi memori sistem operasi Ubuntu.

5. KESIMPULAN

Berdasarkan pengujian dan perbandingan yang telah dilakukan, diperoleh bahwa dengan menggunakan algoritma dan dataset yang sama terdapat perbedaan konsumsi CPU dan memori dari kedua sistem operasi. Konsumsi CPU pada kedua sistem operasi ketika melakukan klasifikasi *malware* menggunakan algoritma *Random Forest* tidak menunjukkan konsistensi linear terhadap jumlah dataset yang digunakan. Persentase konsumsi CPU dapat menurun meskipun dataset yang digunakan lebih banyak daripada dataset sebelumnya. Namun, data hasil pengujian menunjukkan bahwa konsumsi CPU sistem operasi Ubuntu relatif lebih tinggi daripada sistem operasi Kali Linux. Mean konsumsi CPU sistem operasi Ubuntu memiliki selisih sebesar 19,64% lebih tinggi daripada sistem operasi Kali Linux. Sedangkan konsumsi memori pada kedua sistem operasi ketika melakukan klasifikasi *malware* menggunakan algoritma *Random Forest* menunjukkan konsistensi linear terhadap jumlah dataset yang digunakan yaitu konsumsi memori akan meningkat seiring dengan jumlah dataset yang semakin banyak. Data hasil pengujian menunjukkan bahwa konsumsi memori sistem operasi Ubuntu relatif lebih tinggi daripada sistem operasi Kali Linux. Namun, selisih konsumsi memori yang terjadi pada kedua sistem operasi tidak terpaut jauh. Mean konsumsi memori sistem operasi Ubuntu memiliki selisih sebesar 0,06% lebih tinggi daripada sistem operasi Kali Linux. Berdasarkan analisis yang telah dilakukan dapat ditarik kesimpulan bahwa sistem operasi Kali Linux memiliki performa yang lebih baik daripada sistem operasi Ubuntu

dalam hal konsumsi CPU dan memori dalam komputasi klasifikasi *malware* menggunakan teknik dan algoritma *machine learning*.

REFERENSI

- [1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science* (1979), vol. 349, no. 6245, pp. 253–255, Jul. 2015, doi: 10.1126/science.aaa8415.
- [2] M. Molina, "Machine Learning for Sociology" 2019.
- [3] W. Nabi, A. Bansal, and B. Xu, "Applications of artificial intelligence and machine learning approaches in echocardiography," *Echocardiography*, vol. 38, no. 6. Blackwell Publishing Inc., pp. 982–992, Jun. 01, 2021. doi: 10.1111/echo.15048.
- [4] T. Kabbani and F. E. Usta, "Predicting the Stock Trend Using News Sentiment Analysis and Technical Indicators in Spark," Jan. 2022, [Online]. Available: <http://arxiv.org/abs/2201.12283>
- [5] Z. Zhang, H. Al Hamadi, E. Damiani, C. Y. Yeun, and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research."
- [6] N. Milosevic, A. Dehghantanha, and K. K. R. Choo, "Machine learning aided Android malware classification," *Computers and Electrical Engineering*, vol. 61, pp. 266–274, Jul. 2017, doi: 10.1016/j.compeleceng.2017.02.013.
- [7] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "DL-Droid: Deep learning based android malware detection using real devices," *Comput Secur*, vol. 89, Feb. 2020, doi: 10.1016/j.cose.2019.101663.
- [8] M. S. Malik, "Machine Learning in Malware Detection," *LGU International Journal for Electronic Crime Investigation*, vol. 5, no. 3, pp. 29–36, 2021.
- [9] F. A. Kraemer, D. Palma, A. E. Braten, and D. Ammar, "Operationalizing Solar Energy Predictions for Sustainable, Autonomous IoT Device Management," *IEEE Internet Things J*, vol. 7, no. 12, pp. 11803–11814, Dec. 2020, doi: 10.1109/JIOT.2020.3002330.
- [10] S. J. Lee *et al.*, "IMPACT: Impersonation Attack Detection via Edge Computing Using Deep Autoencoder and Feature Abstraction," *IEEE Access*, vol. 8, pp. 65520–65529, 2020, doi: 10.1109/ACCESS.2020.2985089.
- [11] P. Nalmpantis, R. Kalamatianos, K. Kordas, and K. Kermanidis, "Low resources prepositional phrase attachment," in *Proceedings - 14th Panhellenic Conference on Informatics, PCI 2010*, 2010, pp. 78–82. doi: 10.1109/PCI.2010.34.
- [12] K. L. Lai Khine and T. T. Soe Nyunt, "Predictive analytics on high-dimensional big data using principal component regression (PCR)," in *Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering, WCSE 2019 SPRING*, International Workshop on Computer Science and Engineering (WCSE), 2019, pp. 148–153. doi: 10.18178/wcse.2019.03.025.
- [13] X. Zhou, J. Zhang, J. Wan, L. Zhou, Z. Wei, and J. Zhang, "Scheduling-Efficient Framework for Neural Network on Heterogeneous Distributed Systems and Mobile Edge Computing Systems," *IEEE Access*, vol. 7, pp. 171853–171863, 2019, doi: 10.1109/ACCESS.2019.2954897.
- [14] S. Aridhi and E. Mephu Nguifo, "Big Graph Mining: Frameworks and Techniques," *Big Data Research*, vol. 6. Elsevier Inc., pp. 1–10, Dec. 01, 2016. doi: 10.1016/j.bdr.2016.07.002.

- [15] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan, "Toward practical smile detection," *IEEE Trans Pattern Anal Mach Intell*, vol. 31, no. 11, pp. 2106–2111, 2009, doi: 10.1109/TPAMI.2009.42.
- [16] A. Shawahna, S. M. Sait, and A. El-Maleh, "FPGA-Based accelerators of deep learning networks for learning and classification: A review," *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 7823–7859, 2019. doi: 10.1109/ACCESS.2018.2890150.
- [17] C. Y. Mansa Baidoo, W. Yaokumah, and E. Owusu, "Estimating Overhead Performance of Supervised Machine Learning Algorithms for Intrusion Detection," *International Journal of Information Technologies and Systems Approach*, vol. 16, no. 1, Feb. 2023, doi: 10.4018/IJITSA.316889.
- [18] V. R. Dasari, M. S. B. Im, and L. Beshaj, "Solving machine learning optimization problems using quantum computers," 2020. [Online]. Available: <https://www.researchgate.net/publication/336197360>
- [19] T. S. Gunawan, M. K. Lim, N. F. Zulkurnain, and M. Kartiwi, "On the review and setup of security audit using Kali Linux," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 1. Institute of Advanced Engineering and Science, pp. 51–59, Jul. 01, 2018. doi: 10.11591/ijeecs.v11.i1.pp51-59.
- [20] J. Park and Y. I. Eom, "File Fragmentation from the Perspective of I/O Control The 14th ACM Workshop on Hot Topics in Storage and File Systems (HotStorage)."
- [21] B. D. Merrill, A. T. Ward, J. H. Grose, and S. Hope, "Software-based analysis of bacteriophage genomes, physical ends, and packaging strategies," *BMC Genomics*, vol. 17, no. 1, Aug. 2016, doi: 10.1186/s12864-016-3018-2.
- [22] S. Narayan, P. Shang, and N. Fan, "Performance evaluation of IPv4 and IPv6 on Windows Vista and Linux Ubuntu," in *Proceedings - International Conference on Networks Security, Wireless Communications and Trusted Computing, NSWCTC 2009*, 2009, pp. 653–656. doi: 10.1109/NSWCTC.2009.368.
- [23] A. W. Daher, A. Rizik, M. Muselli, H. Chible, and D. D. Caviglia, "Porting rulex software to the raspberry pi for machine learning applications on the edge," *Sensors*, vol. 21, no. 19, Oct. 2021, doi: 10.3390/s21196526.
- [24] F. Kanavati *et al.*, "Weakly-supervised learning for lung carcinoma classification using deep learning," *Sci Rep*, vol. 10, no. 1, Dec. 2020, doi: 10.1038/s41598-020-66333-x.
- [25] Hemavati, V. S. Devi, and R. Aparna, "Multi-label learning by extended multi-tier stacked ensemble method with label correlated feature subset augmentation," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 3, pp. 3384–3397, Jun. 2023, doi: 10.11591/ijece.v13i3.pp3384-3397.
- [26] D. R. Cutler *et al.*, "RANDOM FORESTS FOR CLASSIFICATION IN ECOLOGY," 2007.
- [27] X. Fu, Y. Chen, J. Yan, Y. Chen, and F. Xu, "BGRF: A broad granular random forest algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 44, no. 5, pp. 8103–8117, May 2023, doi: 10.3233/JIFS-223960.
- [28] E. V. A. Sylvester *et al.*, "Applications of random forest feature selection for fine-scale genetic population assignment," *Evol Appl*, vol. 11, no. 2, pp. 153–165, Feb. 2018, doi: 10.1111/eva.12524.
- [29] Y. Chen and R. Zhang, "Research on Credit Card Default Prediction Based on k- Means SMOTE and BP Neural Network," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6618841.

- [30] D. Ghatrehsamani, C. Denninnart, J. Bacik, and M. Amini Salehi, "The Art of CPU-Pinning: Evaluating and Improving the Performance of Virtualization and Containerization Platforms," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Aug. 2020. doi: 10.1145/3404397.3404442.
- [31] S. B. Mishra, "HANDBOOK OF RESEARCH METHODOLOGY," 2017. [Online]. Available: <https://www.researchgate.net/publication/319207471>