Memory Volatile Forensik Untuk Deteksi *Malware* Menggunakan Algoritma *Machine Learning*

http://dx.doi.org/10.28932/jutisi.v4i2.776

Fikri Bahtiar[1], Nur Widiyasono[2], Aldy Putra Aldya[3]

[1,2,3] *Jurusan Teknik Informatika, Fakultas Teknik, Universitas Siliwangi*
Jalan Siliwangi No.24 Tasikmalaya 46155
[1]fikri.bahtiar14@student.unsil.ac.id
[2]nur.widiyasono@unsil.ac.id
[3]aldy@unsil.ac.id

Abstract — **Forensics from volatile memory plays an important role in the investigation of cybercrime. The acquisition of RAM memory or other terms of RAM dump can assist forensic investigators in retrieving much of the information related to crime. Some commonly used tools for analyzing RAM include volatility. It has happened that many forensic investigators are thinking that they probably have malware in the RAM dump. And, if they do exist, they're still not very capable Malware Analysts, so it's hard for them to analyze the possibilities of malware in a RAM dump. The availability of tools such as volatility allows forensic investigators to identify and link the various components to conclude whether the crime was committed using malware or not. user tools such as volatility require command-based knowledge of text and malware analysis. This work is done to assist forensic investigators in detecting and analyzing possible malware from dump RAM. This work is based on the volatility framework and the result is a forensic tool for analyzing RAM dumps and detecting possible malware in it using machine learning algorithms in order to detect offline (not connected to the internet).**

Keywords— **Malware Forensics, Machine Learning, Volatile Memory Analysis, Malware Analysis, Forensic RAM Analysis**

I. PENDAHULUAN

Komputer forensik adalah investigasi dan teknik analisis komputer yang melibatkan tahapan identifikasi, persiapan, ekstraksi, dokumentasi dan interpretasi dari data yang terdapat pada komputer yang berguna sebagai bukti dari peristiwa *cyber crime* [1]. Komputer forensik pada awalnya dilakukan dengan cara menganalisis media penyimpanan dari sebuah sistem yang dicurigai telah terlibat dalam sebuah tindak kejahatan, dimana biasanya sistem perlu dinonaktifkan kemudian dibuat *image* kloning dari media penyimpanan sistem tersebut. *Image* inilah yang dianalisis yang dapat digunakan sebagai barang bukti untuk keperluan investigasi lebih lanjut [2]. Data *volatile* khususnya pada memori fisik atau RAM sebuah *sistem* menggambarkan seluruh kegiatan yang sedang terjadi pada sistem tersebut [2]. Ketersediaan alat seperti Volatilitas memungkinkan penyelidik forensik mengidentifikasi dan menghubungkan berbagai komponen untuk menyimpulkan apakah kejahatan *cyber* itu dilakukan menggunakan *malware* atau tidak. Namun, penggunaan volatilitas membutuhkan pengetahuan tentang alat baris perintah (*Command Line*) serta analisis *malware* statis [2]. Sebagian Alat Forensik yang berfungsi mendeteksi *malware* secara otomatis, tetapi harus selalu terhubung dengan internet, dan deteksi *malware* yang dilakukan terbatas. Pekerjaan yang disebutkan dalam makalah ini terinspirasi untuk menerapkan algoritma *machine learning* dan otomatisasi langkah – langkah dasar. Keuntungan terbesar dari alat ini adalah, Pengguna dapat mendeteksi semua proses yang berjalan pada *memory volatile* dan tidak harus terkoneksi dengan internet dan Pengguna tidak perlu mengingat perintah, sintaknya atau bahkan ketika mau menggunakan perintah mana. Ini sangat berguna bagi mereka yang tidak lebih suka bekerja pada utilitas baris perintah karena mereka menghindari mengingat perintah. Solusi yang diusulkan disebut *Memory Volatile* Forensik untuk deteksi *malware* menggunakan algoritma *machine learning*, adalah solusi untuk pengguna yang ramah dan akurat untuk mengatasi masalah diatas, juga menganalisis dan memberikan laporan akhir yang akurat.

II. LANDASAN TEORI

A. Digital Forensik

Digital Forensik merupakan bagian dari ilmu forensik yang melingkupi penemuan dan investigasi materi (data) yang ditemukan pada perangkat digital. Penguasaan ilmu ini tidak hanya ditunjukan kepada kemampuan teknis semata tetapi juga terkait dengan bidang lain, seperti bidang hokum [3].

B. Ilmu Forensik

Forensik adalah multi disiplin ilmu yang digunakan untuk tujuan hukum dengan tidak memihak pada bukti ilmiah

untuk digunakan dalam pengadilan hukum, dalam penyelidikan dan pengadilan pidana [4].

C. Forensik Jaringan

Forensik Jaringan Merupakan ilmu keamanan komputer berkaitan dengan investigasi untuk menemukan sumber serangan pada jaringan berdasarkan bukti log, mengidentifikasi, menganalisis, serta merekonstruksi ulang kejadian tersebut. Istilah *Network Forensik* memang di ambil dari terminologi yang berhubungan dengan kriminologi [5].

D. Cybercrime

Cyber Crime adalah segala macam penggunaan jaringan computer untuk tujuan kriminal dan atau kriminal berteknologi tinggi dengan menyalahgunakan kemudahan teknologi digital [6].

E. Barang Bukti

Barang Bukti (*Evidence*) yang diartikan pada forensic tidak lain ialah informasi dan data dari apa yang didapatkan pada suatu kasus. Barang bukti bagian terpenting dalam sebuah kasus kejahatan untuk memecahkan kasus tersebut [7]

1) Barang bukti elektronik: Barang bukti ini bersifat fisik dan dapat di kenali secara visual, sehingga investigator dan analis forensic harus sudah memahami serta mengenali masing-masing barang bukti elektronik ini ketika sedang melakukan proses pencarian (*searching*) barang bukti di TKP. Jenis barang bukti elektronik ini antara lain :

- PC
- *Notebook*
- *Tablet*
- *Handphone*
- *Flashdisk*
- *Harddisk*
- *CD/DVD*
- *Router,Switch*
- *Kamera*
- *CCTV*

2) Barang bukti digital: Barang bukti digital sangat rentan terhadap perubahan informasi didalamnya, perlu penanganan untuk menjaga keutuhan barang bukti.

- *Logical Drive*
- *Deleted File / Unallocated Custer*
- *Lost File*
- *Slack File*
- *Log File*
- *Encrypted File*
- *Steganography File*
- *Office File*
- *Audio File*
- *Image File*
- *Video File*
- *Email / Electronic Mail*
- *User ID and Password*
- *SMS / Short Message Service*
- *Call Log*

F. Memory Volatile

Memori volatile adalah penyimpanan komputer yang hanya menyimpan datanya saat perangkat diaktifkan.
Sebagian besar RAM (*Random Access Memory*) yang digunakan untuk penyimpanan primer di komputer adalah memori yang mudah menguap. RAM jauh lebih cepat untuk dibaca dan ditulis dibandingkan dengan jenis penyimpanan lain di komputer, seperti hard disk atau media yang dapat dipindahkan. Namun, data dalam RAM tetap ada ketika komputer sedang berjalan, akan tetapi sebaliknya ketika komputer dimatikan, RAM kehilangan datanya. [8]

G. Machine Learning

Machine learning (Pembelajaran Mesin) merupakan kemampuan komputer untuk melakukan pembelajaran tanpa harus menjelaskan (*programmed*) secara eksplisit kepada komputer [9], atau menurut [21] suatu komputer dikatakan melakukan pembelajaran dari pengalaman (E) terhadap tugas (T) dan mengukur peningkatan kinerja (P), jika kinerja Tugas (T) diukur oleh kinerja (P), meningkatkan pengalaman (E) .

III. PENELITIAN TERKAIT

Investigator Forensik, Analis *Malware*, dan perusahaan sedang bekerja dari bertahun-tahun dalam mengotomatisasi proses analisis untuk memudahkan pekerjaan mereka. Kecuali aspek forensik seperti yang dibahas dalam hal ini, ada banyak tools sandBoxes tersedia yang gratis seperti Cuckoo [10] yang menyediakan otomatisasi dalam proses analisis *malware*. Juga Michael Bailey, et. Al. mengusulkan otomatisasi klasifikasi dan analisis cara kerja *Malware* Internet [11]. Manuel Egele, et. Al. dalam survei mereka mendiskusikan berbagai Tools sanBoxes dan alat analisis *malware* otomatis [12]. Akan tetapi hasil otomatisasi *memory forensic* sangat sedikit dalam proses pengerjaannya sampai selesai Tomer Teller, et. Al. mengusulkan solusi berdasarkan cuckoo, Volatility dan IDA [13] dalam jurnal mereka di Blackhat [14], tetapi, itu sangat tergantung pada tool Cuckoo. Penelitian yang lain seperti Logen, Höfken dan Schuba menyediakan solusi berbasis GUI sebagai Perkembangan Volatilitas dalam jurnal mereka [15], meskipun pekerjaan yang diajukan oleh mereka melakukan beberapa tugas dasar secara otomatis, Sementara alat lain eVOLe yang dikembangkan oleh James Habben [16] adalah alat berbasis web, akan tetapi alat itu hanya menanyakan profil Image pada saat eksekusi, yang menunjukkan bahwa pengguna diminta untuk menjalankan Volatilitas secara terpisah untuk mendapatkan profil dan pekerjaan menjadi

membosankan. GUI eVOLe yang lebih lanjut tidak menyediakan Analisis *Malware* yang lengkap. Rughani Vimal, et. Al. mengusulkan solusi GUI untuk proses deteksi *malware* secara otomatis, akan tetapi dalam pendekteksian *malware* dengan jumlah yang terbatas dan harus terhubung dengan internet [17]. Untuk mengatasi semua masalah seperti itu, menggunakan *Memory Volatile Forensic*. Untuk deteksi *malware* menggunakan Algoritma *Machine Learning* [18] diusulkan di bagian berikutnya.

IV. ALAT YANG DIUSULKAN

Penelitian ini dilakukan untuk membantu Digital Forensic Investigator, diasumsikan yang tidak ahli dalam menganalis *malware* tetapi diperlukan untuk memiliki beberapa mekanisme yang dapat dengan mudah mereka identifikasi kehadiran *malware* apa pun dalam dump *memory* RAM. Alat yang diusulkan yaitu *Memory Volatile Forensic* menggunakan Algoritma *Machine Learning*. Aplikasi ini berbasis desktop untuk melakukan *memory forensic* secara otomatis terkait pekerjaan yang rumit dan membosankan. GUI dari alat ini dikembangkan dalam bahasa pemrograman Java [19], berbasis desktop, untuk melakukan forensik memori. Para investigator forensik harus melakukan langkah-langkah yang sangat minimal dalam menganalisis laporan dari *memory forensic*. Alat yang dibuat ini hanya membutuhkan satu *file image dump* RAM yang akan dianalisis. Alat ini menyediakan fitur untuk mengunggah *image dump* RAM. Setelah file image diunggah dan mulai mengeksekusi secara komplek proses yang berbeda secara otomatis di latar belakang, pengguna juga akan mendapatkan tampilan secara langsung pada status proses yang sedang dieksekusi. Proses otomatis pertama yang dilakukan oleh alat ini adalah mengekstrak file dump dan menempatkan file dump yang *valid* di tempat yang tepat untuk dieksekusi lebih lanjut. Langkah selanjutnya adalah memilih profil *sistem operasi* yang disediakan oleh alat ini. Setelah memberikan informasi file image dan profil, alat ini mulai menjelajahi dan menganalisis dump proses pada *memory volatile*, proses ini berperan penting dalam mengidentifikasi serangan *malware*. Sebagian besar *malware* termasuk *ransomware* berbasis jaringan dan berfungsi sebagai *botnet*. Sebagian besar *malware* ini harus terhubung ke pusat kontrol untuk mengeksekusi perintah selanjutnya atau mengirim informasi penting atau rahasia. Untuk menyelesaikan komunikasi semacam itu, *malware* menggunakan IP dengan port yang terbuka. [20] Menyatakan "Kita harus mengakses memori fisik komputer sistem untuk menemukan informasi yang lebih penting, seperti alamat IP penyusup, informasi tentang program jahat yang sedang berjalan, proses, worm, Trojan dan sebagainya pada jurnal mereka [20]. Untuk mengidentifikasi IP dengan Port yang terbuka seperti itu, Alat ini akan menganalisis Koneksi Jaringan dari file dump RAM yang diberikan. Ini akan memberikan semua informasi yang mungkin dan perlu secara detail kepada Anda untuk mengidentifikasi IP atau port yang terbuka. Jadi jika ada IP atau port yang ditemukan, kita dapat dengan mudah menghubungkan dengan proses yang terkait. Penting untuk dicatat bahwa yang di sebutkan di atas proses mungkin menjadi sulit bagi peneliti Forensik yang awam, jika mereka melakukan pemeriksaan *malware* secara manual untuk setiap IP, port dan proses. Alat ini yang berperan sangat penting dalam melakukan proses yang disebutkan di atas secara otomatis. Setelah mengidentifikasi Proses yang dicurigai *malware*, tugas penting berikutnya adalah untuk menggali lebih banyak tentang proses itu dan mengidentifikasi executable, DLL, dan file yang digunakan oleh Proses itu. Alat ini menyediakan fitur pengunduhan untuk diproses, jadi para forensic *investigators* bisa bereksperimen dengan mencurigai file *executable* atau proses di lingkungan yang terisolasi atau mereka bisa mengirim file tersebut untuk diajukan ke pusat penelitian *malware* untuk penyelidikan lebih lanjut. Selain itu untuk sistem *windows Registry* adalah sumber untuk forensic artefak yang dapat digunakan selama investigasi, insiden penanganan respons, dan analisis *malware*. Selain fitur yang disebutkan di atas, Alat ini menyediakan fitur pemindaian file proses individual untuk virus, *worm, Trojans* dan segala macam *malware*. Alat ini menggunakan algoritma *Machine Learning* sebagai proses pendeteksian *malware* untuk dapat mendeteksi *malware* secara *offline* dan tidak terbatas selama proses berjalan. Alat ini adalah wadah untuk menutupi dan mengotomatiskan semua langkah yang diperlukan oleh proses memori forensik dalam membantu *digital forensic investigator*. Pengguna akan mendapatkan hasil yang akurat tanpa mengetahui baris perintah yang diberikan *tools volatility*. Alat ini tersedia hanya untuk W*indows* dan mendukung file *image dump* memori dari Windows, Linux dan Mac.

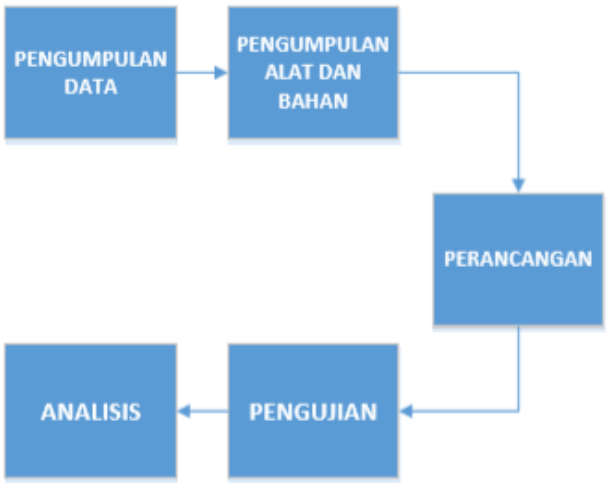
V. METODOLOGI

A. Tahapan Penelitian

Merupakan langkah-langkah dalam melakukan penelitian, berikut tahapan penelitian yang dilakukan dapat dilihat pada gambar 1 :

1) *Pengumpulan data:* Pengumpulan informasi dari sumber yang berkaitan dengan penelitian, studi literatur yaitu sumber-sumber dari jurnal, buku, internet, artikel dan lain-lain.
2) *Pengumpulan Alat dan Bahan:* Pengumpulan kebutuhan-kebutuhan yang digunakan pada penelitian, baik berupa perangkat keras, dan perangkat lunak yang mendukung dalam pembuatan alat forensik ini.
3) *Perancangan:* Membuat rancangan dan interface untuk menghubungkan framework volatilitas dengan algoritma *machine learning*.

4) *Pengujian:* Pengujian hasil analisis *dump memory*, dan proses deteksi *malware* menggunakan algoritma *machine learning*.
5) *Analisis:* yaitu untuk menganalisis terhadap proses yang terinfeksi *malware* pada *memory volatile.*



Gambar 1. Tahapan Penelitian

B. Kebutuhan Bahan

1) Perangkat Keras

Beberapa perangkat keras yang dibutuhkan dalam pembuatan *tool forensic* ini, berikut dijelaskan pada Tabel I.

TABEL I

PERANGKAT KERAS YANG DIBUTUHKAN

NO	NAMA	JUMLAH
1.	LAPTOP *HP* INTEL CORE I3 RAM 4GB	1 UNIT
2.	FLASHDISK	1 UNIT

2) Perangkat Lunak

Beberapa perangkat lunak yang dibutuhkan untuk membuat *tools forensic* ini, berikut dijelaskan pada Tabel II.

TABEL II

PERANGKAT LUNAK YANG DIBUTUHKAN

NO	NAMA	FUNGSI
1.	WINDOWS 7, 8,1, 10	SISTEM OPERASI
2.	FRAMEWORK VOLATILITY	ANALISIS MEMORY
3.	SCIKIT-LEARN	LIBRARY PYTHON UNTUK ALGORITMA MACHINE LEARNING
4.	PEFILE	LIBRARY PYTHON UNTUK MEMBACA PE-HEADER
5.	JAVA NETBEANS	PROGRAM GUI YANG MENGHUBUNGKAN FRAMEWORK VOLATILITY DENGAN ALGORITMA MACHINE LEARNING.

C. Perancangan

1) Proses Permodelan Algortima Klasifikasi

Algoritma Klasifikasi yang digunakan adalah dengan cara mengevaluasi 5 algoritma yaitu :

1. *Algoritma Naive Bayes*

Naïve Bayes adalah salah satu model paling praktis dalam algoritma *machine learning*. Mitchell memperkenalkan metode Naive Bayes secara rinci dalam buku nya [21]. Michie, Spiegelhalter, dkk meneliti dan mendalami model Naive Bayes [22], dan mereka membandingkan algoritma model klasifikai Naive bayes dengan algoritma pembelajaran lainnya, seperti pohon keputusan (Decision Tree). Hasil studi mereka menunjukkan bahwa di Indonesia kebanyakan kasus kinerja Naive Bayes sama baiknya dengan model lainnya.

2. *Algoritma Decision Tree / Pohon Keputusan*

Metode pohon keputusan mencapai popularitasnya karena kesederhanaannya. Bisa mengoptimalkan dengan baik dengan dataset yang besar dan dapat menangani ketidakakuratan di dataset dengan sangat baik. Keuntungan lainnya adalah tidak seperti algoritme lain, seperti SVM atau KNN, pohon keputusan beroperasi dalam "White Box", yang berarti bahwa kita dapat melihat dengan jelas bagaimana hasil yang diperoleh dan keputusan mana yang paling akurat. Fakta-fakta ini menjadikannya solusi untuk diagnosis medis, filtering spam, filtering keamanan, dan bidang lain. [21].

3. *Algoritma Random forest*

Aloritma Random Forest merupakan algoritma yang dapat dipakai pada klasifikasi dan regresi. Diperkenalkan oleh Leo Breinan [23] dimana teknik ini dapat menghasilkan banyak pohon klasifikasi. Bagaimana Random Forest dapat menghasilkan klasifikasi berawal mula dari input vector yang bergerak menuruni masing masing pohon. Masing-masing pohon merupakan klasifikasi berdasarkan mekanisme suara terbanyak atau vote untuk menandai class tersebut. Sehingga pepohonan (forest) dapat menentukan klasifikasi berdasarkan hasil voting tersebut.

4. *Algoritma AdaBoost*

Algoritma AdaBoost adalah algoritma klasifikasi yang dapat meningkatkan algoritma *machine learning* yang lemah dengan akurasi sedikit lebih baik dari pada menebak secara acak menjadi algoritma *machine learning* yang kuat. [24].

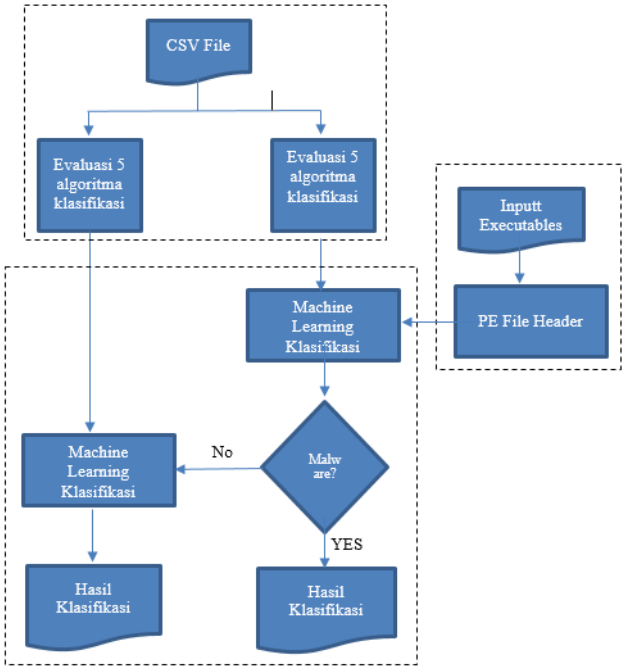
5. *Gradient bossting*

Algoritma Gradient boosting seperti halnya keluarga algoritma Boosted lainnya memiliki kemampuan untuk meningkatkan akurasi prediktif model. Beberapa algoritma boosting lainnya seperti: XGBoost, AdaBoost dan GradientBoost memiliki formula matematika tersendiri dan bervariasi. Konsep Gradient Boosting terletak pada pengengembangannya yang mana memiliki ekspansi tambahan terhadap fitting criterion [25].

Sehingga dapat efektif dalam mengklasifikasi file PE Header pada dataset dengan hasil ektraksi file PE yang

p-ISSN : 2443-2210
e-ISSN : 2443-2229

diinput dari modul python yaitu "pefile". Berikut pada gambar 2 adalah proses permodelan algoritma klasifikasi :



Gambar 2. Proses Permodelan algortima klasifikasi

2) Pengambilan sampel DataSet

DataSet yang dikumpulkan yaitu sebanyak 41,323 file sampel dari hasil ekstraksi folder system32 di Windows 7,8.1 dan 10, File-file dalam folder system32 diekstrak setelah instalasi OS dengan update terbaru. Sampel *malware* yang dapat dieksekusi diunduh dari situs [26]. Jumlah total sampel *malware* adalah 96.724 yang mengandung PE-header. "pefile" yang merupakan salah satu dari modul python dipilih untuk mendeteksi PE-header pada suatu file dan mengekstraksi informasi header-PE dari file PE tersebut [27]. Sehingga total jumlah didalam sampel yaitu 138.047

Berikut pada tabel III adalah Atribut-atribut atau *parameter* yang berhubungan dengan dataset sebanyak 57 atribut.

TABEL III

ATRIBUT ATAU PARAMETER

ATRIBUT / *PARAMETER* YANG DIGUNAKAN				
Name	AddressOfEntryPoint	MajorSubsystemVersion	SizeOfHeapCommit	ResourcesMaxEntropy
md5	BaseOfCode	MinorSubsystemVersion	LoaderFlags	ResourcesMeanSize
Machine	BaseOfData	SizeOfImage	NumberOfRvaAndSizes	ResourcesMinSize
SizeOfOptionalHeader	ImageBase	SizeOfHeaders	SectionsNb	ResourcesMaxSize
Characteristics	SectionAlignment	CheckSum	SectionsMeanEntropy	LoadConfigurationSize
MajorLinkerVersion	FileAlignment	Subsystem	SectionsMinEntropy	VersionInformationSize
MinorLinkerVersion	MajorOperatingSystemVersion	DllCharacteristics	SectionsMaxEntropy	legitimate
SizeOfCode	MinorOperatingSystemVersion	SizeOfStackReserve	SectionsMeanRawsize	ResourcesMaxEntropy
SizeOfInitializedData	MajorImageVersion	SizeOfStackCommit	SectionsMinRawsize	ResourcesMeanSize

Atribut / *Parameter* diatas menunjukan bahwa setiap file *Portable Executable (PE)* yang diekstrak akan diklassifikasikan dengan 57 atribut yang sebelumnya sudah dibuat

VI. HASIL DAN PEMBAHASAN

Dari 5 algoritma yang digunakan dapat diambil salah satu algoritma yang unggul dalam mengklasifikasi *malware* didalam dataset, hasil dari pengetesan data *training* nantinya akan digunakan untuk melatih algoritma untuk mencari model yang cocok. Berikut adalah hasil dari pengetesan ke 5 algoritma dengan menggunakan dataset pada Tabel IV:

TABEL IV

PENGUJIAN ALGORITMA

NO	**ALGORITMA**	**JUMLAH DATASET**	**HASIL PENGETESAN**
1.	GNB	138.047	69.952916 %
2.	DECISION TREE	138,047	98.913437 %
3.	RANDOM FOREST	138.047	99.406012 %
4.	ADAPTIVE BOSS	138.047	98.540384 %
5.	GRADIENT BOOSTING	138.047	98.790293 %

Berikut pada tabel V adalah hasil pengujian dengan berbagai macam sampel *image* Windows dengan fitur yang disediakan.

TABEL V

PENGUJIAN DENGAN BERBAGAI MACAM SAMPEL PROFIL SISTEM OPERASI

FEATURE	**PROFILE SISTEM OPERASI**					
	Win XPSP1x86	Win XPSP2x86	Win 7SP1x86	Win 7SP1x86	Win 10x64_14939	Win10x86_14939
PROCESS LIST	v	v	v	v	v	v
DLL	v	v	v	v	v	v

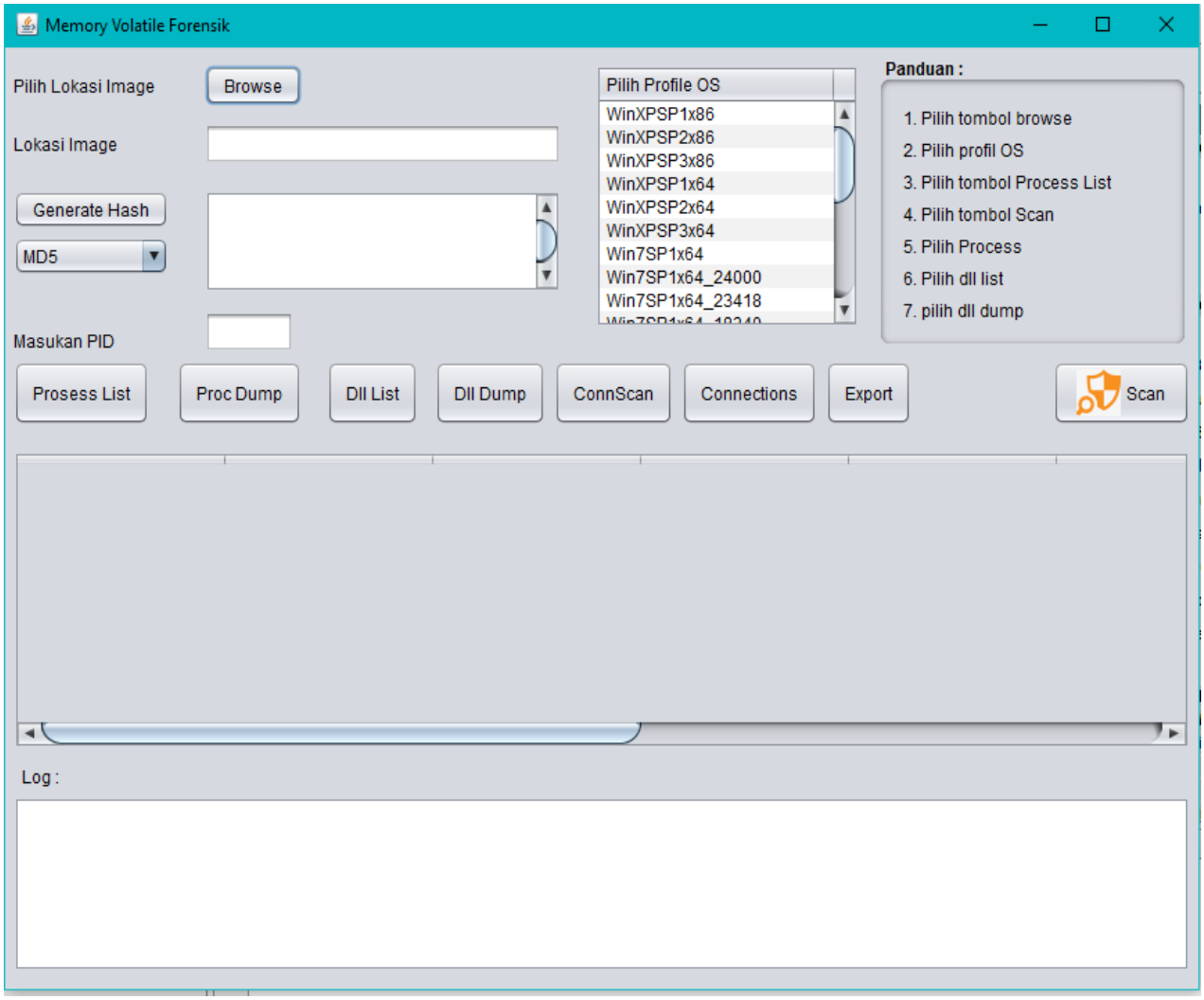
LIST						
CONN SCAN	v	v	x	x	x	x
CONNECTIONS	v	v	x	x	x	x
PROC DUMP	v	v	v	v	v	v
SCAN MALWARE	v	v	v	v	v	v

Ket : v = berjalan x = tidak berjalan

Setelah mendapatkan file, seseorang dapat menginstal alat ini setelah terinstal *software* pendukung yaitu :

a. JDK & JRE versi 1.8.0
b. Python 2.7
c. Pandas
d. Numpy
e. Pickle
f. Scipy
g. Scikit-learn
h. Pefile

Setelah terinstal semua, alat ini bisa dijalankan. Gambar 3 adalah tampilan detail dengan fitur-fitur penting.



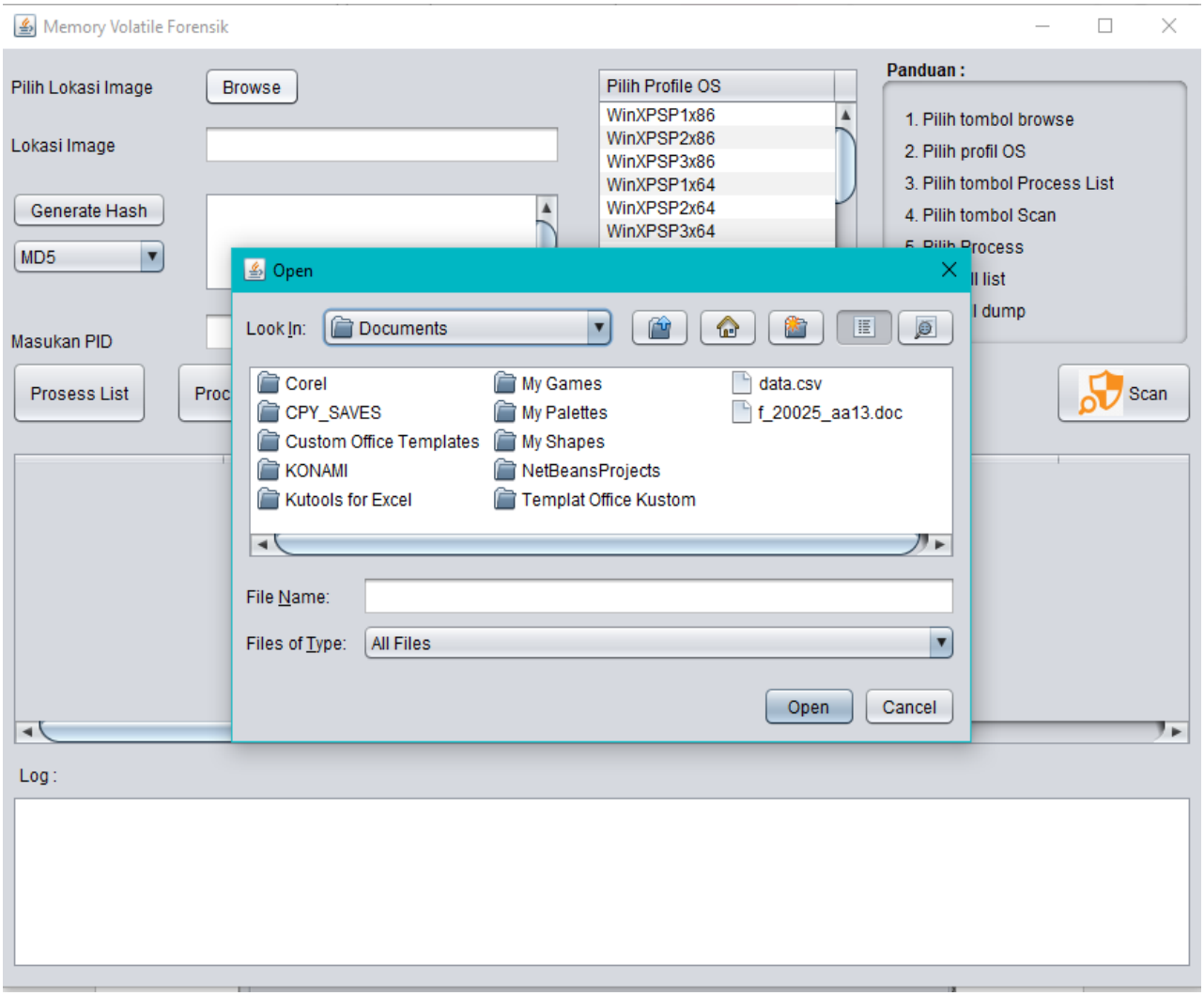
Gambar 3. Tampilan detail dengan fitur-fitur penting

Gambar 3 diatas memiliki fitur-fitur penting yaitu:

1. Browse yaitu untuk mengambil image yang akan dianalasis.
2. Profil OS yaitu jenis Profil yang terdapat pada file image.
3. Generate Hash yaitu untuk mengetahui jenis dan nilai hash pada file image yang dimasukan.
4. Proccess List yaitu untuk mengidentifikasi process yang berjalan.
5. Dll list yaitu untuk mengidentifikasi letak dll yang terhubung dengan process.

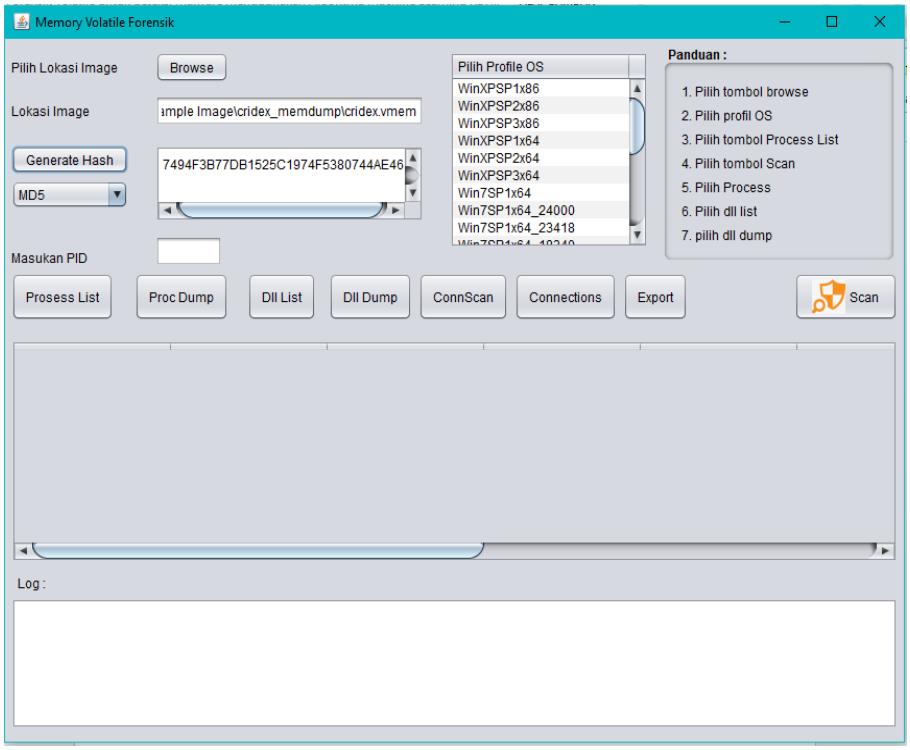
p-ISSN : 2443-2210
e-ISSN : 2443-2229

6. Connscan yaitu mengetahui remote IP yang digunakan.
7. Cconnections yaitu untuk mengetahui jenis nomor IP dan port yang digunakan.
8. Proc Dump yaitu untuk mengambil salah satu process / dll pada memory dan dapat diletakan pada local drive.
9. Scan yaitu untuk mengetahui process yang terinfeksi malware.



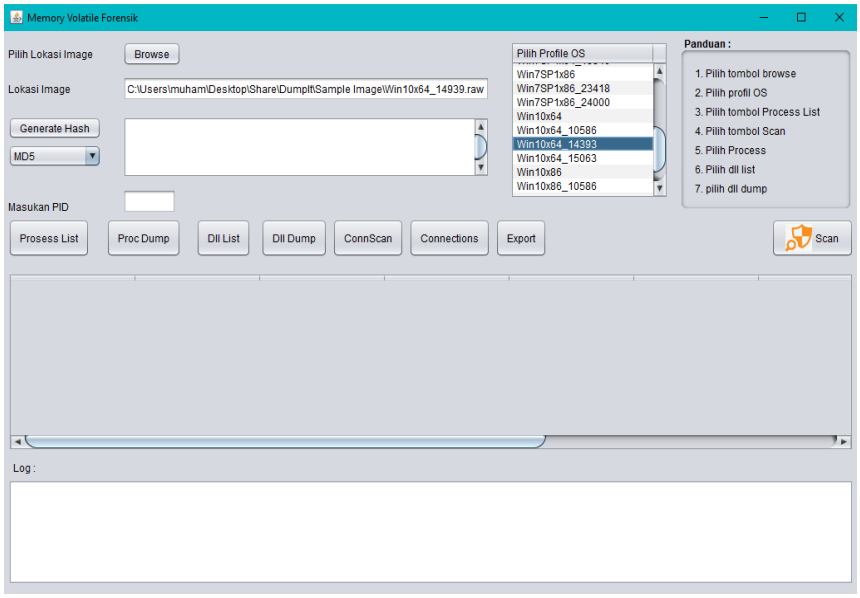
Gambar 4. Memasukan *file dump* RAM

Dari gambar 4, pengguna dapat mengunggah file *image dump memory* menggunakan tombol *browse*. Karena ukuran *dump memory* mungkin dalam GB atau MB sesuai ukuran memory fisik yang diakusisi, pengguna diminta untuk mengklik tombol *Analyze Dump*, Alat ini akan mulai menganalisis *file dump* tersebut dan akan memperbarui menampilkan hasil dari forensik memori berupa profil proses yang sedang berjalan.



Gambar 5. Menampilkan Nilai Hash pada file image yang diinputkan

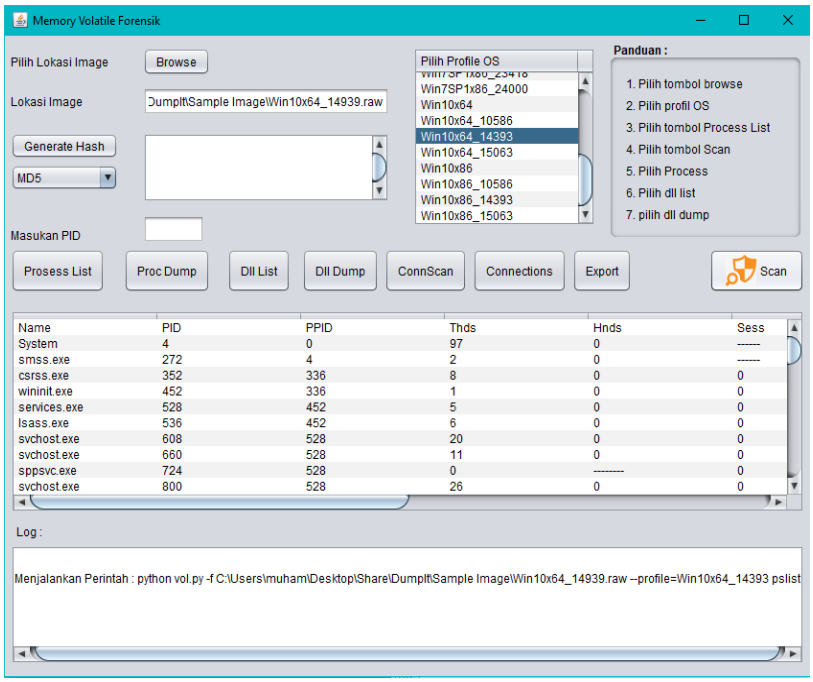
Dari Gambar 5, setelah memasukan file image pengguna juga dapat melihat nilai hash suatu file, dengan cara memilih tombol *generate hash*. Nilai hash ini berfungsi sebagai keaslian suatu data, jika file image tersebut sudah ada yang memodifikasi, dapat diketahui pada bagian hash nya akan berubah, tidak akan sama dengan yang original.



Gambar 6. Memilih profil sistem operasi

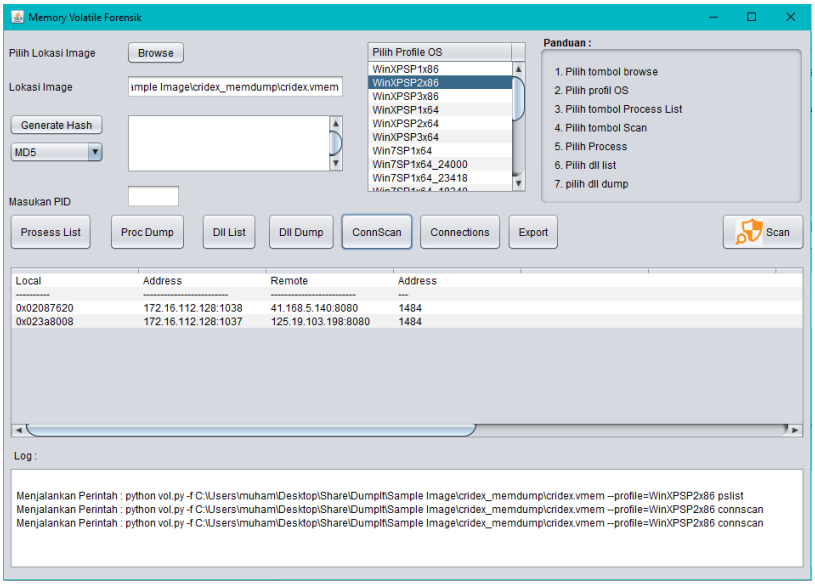
Dari Gambar 6, pengguna dapat memilih Profile Sistem Operasi sesuai dengan image yang diinputkan yang berfungsi menampilkan sistem operasi yang digunakan oleh file image tersebut, dari mulai sistem operasi windows , linux dan MacOS.

p-ISSN : 2443-2210
e-ISSN : 2443-2229



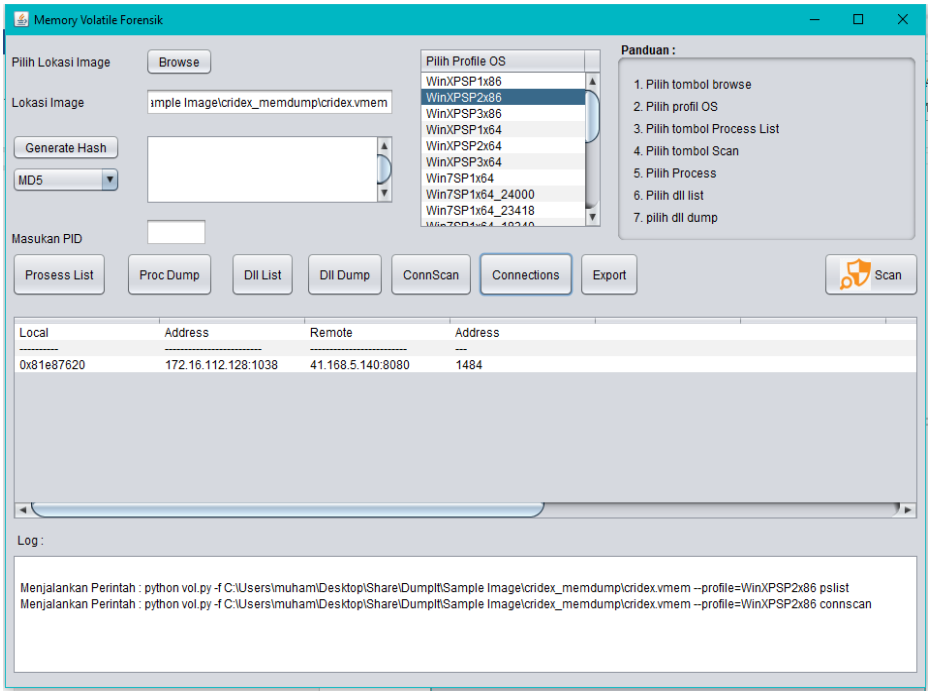
Gambar 7. Informasi dan proses keseluruhan dari *memory volatile*

Dari Gambar 7, setelah analisis berhasil alat ini mengalihkan pengguna ke tombol yang lain di *interface* awal untuk menampilkan daftar proses dengan detail yang relevan seperti *thread, handle*, dan hal lain terkait dengan setiap proses.



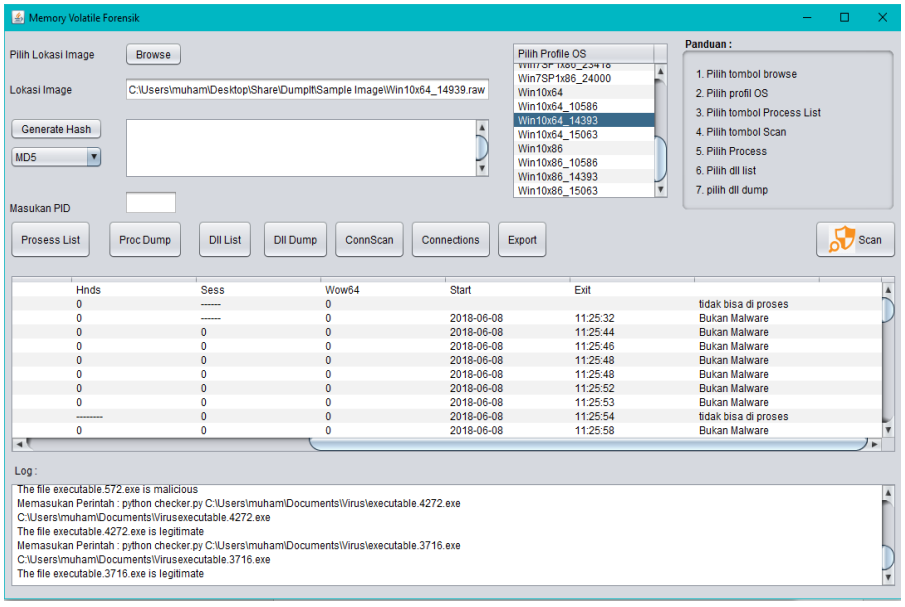
Gambar 8. Tampilan *ConnScan* pada jaringan

Dari Gambar 8, fungsi Connscan yaitu *Scanning Memory* untuk mendapatkan koneksi *TCP*, termasuk *port* yang tertutup atau tidak terhubung *(Unlinked)* Prosess ini hanya berlaku pada file Windows XP atau 2003. Untuk indentifikasi remote ip tentu kita harus mengecek koneksi layaknya kita menggunakan Netstat, plugin yang digunakan untuk proses ini, kita bisa gunakan connections untuk melihat koneksi yang terbuka serta connscan untuk melihat koneksi TCP.



Gambar 9. Tampilan *Connection* pada jaringan

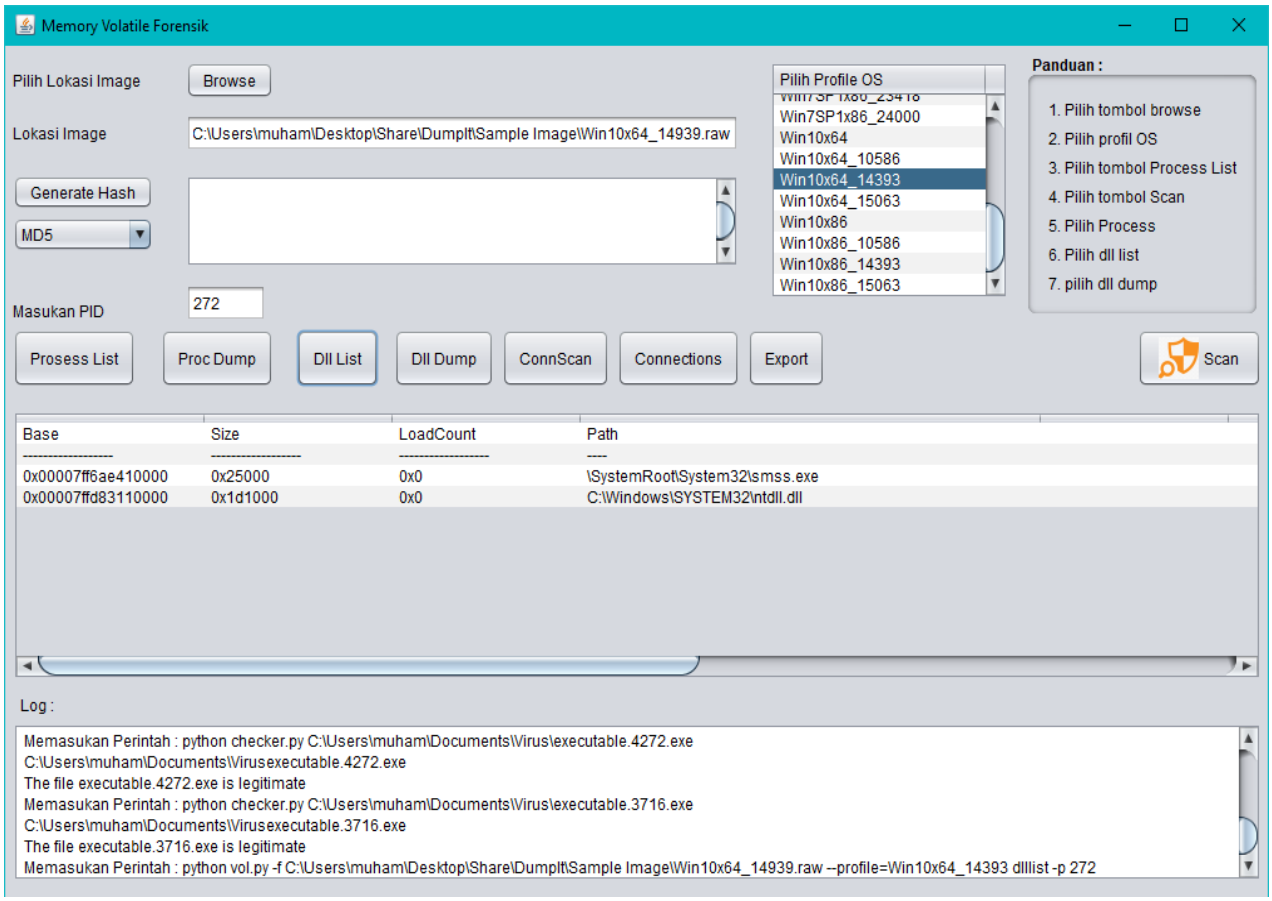
Dari Gambar 9, pengguna dapat mengetahui IP dan port yang terhubung pada komputer. Sehingga dapat diketahui port mana yang digunakan oleh *malware* untuk menyerang sistem, akan tetapi *tools* ini dapat menggunakan menu connection dan connscan jikalau tipe profilnya menggunakan WinXPSP1x86/64 dan WinXPSP2x86/64.



Gambar 10. Proses *scan file executable* yang dicurigai *malware*

Dari Gambar 10, perspektif dalam menganalisis *malware* ini adalah proses yang mencurigakan dapat dianalisis dengan mengklik tombol scan. Itu akan memindai proses individu untuk virus, worm, Trojan. Ketika Anda mengklik tombol scan, itu akan menjalankan algorima *machine learning* untuk mendeteksi *malware* pada proses tersebut.

p-ISSN : 2443-2210
e-ISSN : 2443-2229



Gambar 11. Menampilkan dll list dari proses yang dipilih

Dari Gambar 11, pada proses ini pengguna dapat mengetahui file-file terinfeksi yang digunakan oleh *malware* dalam menyerang sistem operasi, dan dapat diekstrak melalui tombol *procdump*.

VII. KESIMPULAN

Hasil dari pekerjaan ini dapat disimpulkan bahwa dari ke 5 algoritma tersebut, algoritma *Random Forest* yang unggul dalam pengklasifikasian *malware* menggunakan dataset. Dan akan memberikan manfaat dan bantuan untuk penyelidik forensik dalam menganalisis memori *volatile* dan mendeteksi *malware* secara offline (tidak terhubung ke internet) yang mungkin ada pada *memory volatile*. Alat ini telah diuji dengan berbagai sampel *image* dan memberikan hasil yang akurat menurut dataset yang digunakan yaitu berisi file system32 dan *malware*. Akurasi dan keramahan pengguna alat ini akan membantu penyidik *forensic investigator* dan alat ini juga dapat mengurangi biaya pelatihan penyelidik forensik untuk menganalisis *malware*.

Padahal, alat ini memenuhi semua persyaratan dalam menganalisis *malware* dengan algoritma klasifikasi, masih ada ruang lingkup untuk pengembangan. Alat ini dapat diperluas dalam menganalisis *malware* secara detail. Alat yang dikerjakan saat ini hanya mendukung proses yang mengandung / terinfeksi *malware*, sehingga tidak dapat memberikan informasi yang lebih detail dari *malware* tersebut. Terakhir, dibuatkan sebuah GUI untuk proses otomatis ini dapat di operasikan pada sistem operasi yang lain seperti Linux dan Mac untuk lebih bervariasi.

DAFTAR PUSTAKA

[1] Michael Solomon, Diane Barrett, Neil Broom, *Computer Forensics Jumpstart*, Alameda: SYBEX Inc,2005.

[2] A. Frank, "Live Forensics – Diagnosing Your System Without Killing It First," *Communication of The ACM.*, vol. 49, pp. 63-66, Feb. 2006.

[3] B. Raharjo, "Sekilas Mengenai Forensik Digital," *Jurnal Sosioteknologi.*, vol.29, pp. 384-287, Agu. 2013.

[4] E. S. Wijaya and Y. P, "Integrasi Metode Steganografi DCS pada Image dengan Kriptografi Blowfish sebagai Model Anti Forensik untuk Keamanan Ganda Konten Digital," *Prosiding Nasional Aplikasi Teknologi Informasi (SNATi)*, 2015, paper 1907 – 5022, p. 11-17.

[5] R. U. Putri and J. E. Istiyanto, "Analisis Forensik Jaringan Studi Kasus Serangan SQL Injection pada Server Universitas Gadjah Mada," *Indonesian Journal of Computing and Cybernetics Systems*., vol. 6, pp. 101-112, Jul. 2012.

[6] M. N. Al-Azhar, *Digital Forensic : Panduan Praktis Investigasi Komputer*, Jakarta: Salemba Infotek, 2012.

[7] K. E. Pramudita, "Brute Force Attack dan Penerapannya pada Password Cracking," *Prosiding makalah IF3051 Strategi Algoritma*, 2011.
[8] Rouse Margaret. (2018) Home/Topic/Data Center/Storage Hardware/volatile memory. [Online]. Tersedia: https://whatis.techtarget.com/definition/volatile-memory
[9] A.L.Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *in IBM Journal.*, vol 3, pp. 210-229, july. 1959.
[10] Cuckoo Sandbox. (2017). A malware Analysis system. [Online].Tersedia https://www.cuckoosandbox.org
[11] B. Michael, et.all, "Automated classification and analysis of internet malware," Prosiding international conference on Recent advances in intrusion detection, 2007, paper 3-540-74319-7, p. 178-197.
[12] Manuel Egele, Theodoor Scholte, Engin Kirda and Christopher Kruegel, "A Survey on Automated Dynamic Malware Analysis Techniques and Tools," *in the ACM Computing Surveys*, Vol.44 Issue 2, Article No. 6, pp. 1–49, 2012.
[13] IDA. (2017) Multi-processor disassembler and debugger. [Online]. Tersedia: https://www.hex-rays.com/products/ida/
[14] Tomer Teller, Adi Hayon, Enhancing Automated Malware Analysis Machines with Memory Analysis, Blackhat Arsenal, pp. 1-5, 2014.
[15] L. Steffen, Hans Hofken, Marko Schuba, "Simplifying RAM Forensics A GUI and Extensions for the Volatility Framework," *Prosiding Seventh International Conference on Availability, Reliability and Security*, 2012, paper 978-1-4673-2244-7, p. 620-624.
[16] (2017) eVOLve by JamesHabben. [Online]. Tersedia : https://github.com/ James Habben/evolve
[17] Rughani Vimal, Rughani Parag H, "AUMFOR : Automated Memory Forensics for Malware Analysis," *in Asian Journal of Engineering And Applied Technology*, Vol.6, No.2, pp.36-39, 2017
[18] K. Dong-Hee, W. Sang-Uk, L. Dong-Kyu, C. Tai-Myoung, "Static Detection of Malware and Benign Executable Using Machine Learning Algorithm," *Prosiding in The Eighth International Conference on Envolving*, 2016, paper 978-1-61208-516-6, p. 14-9.
[19] (2018) Oracle. [Online]. Tersedia:http://www.oracle.com/technetwork/articles/java/index-137868.html
[20] Liming Cai, Jing Sha ,Wei Qian, "Study on Forensic Analysis of Physical Memory", *Prosiding of 2nd International Symposium on Computer, Communication, Control and Automation*, 2013, paper 978-90786-77-91-8, p. 221-224.
[21] Tom Mitchell, *Machine Learning 1 Edition*, New York: McGraw Hill, March, 1997: 112-143.
[22] Donald Michie, David J. Spiegelhalter, Charles C. Taylor. "Machine Learning, Neural and Statistical Classification," *USA: Ellis Horwood, NJ*, 1994.
[23] L. Breiman, "Random Forests," *Kluwer Academic Publishers.*, vol. 45, pp. 5-32, Oct. 2001.
[24] Ying CAO, Qi-Guang MIAO, Jia-Chen LIU, Lin GAO, "Advance and Prospects of AdaBoost Algorithm," *Acta Automatika Sinica.*, Vol. 39, pp. 745-758, June. 2013.
[25] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Ann. Stat.*, vol. 29, p. 5, 2001.
[26] Vxheaven (2016). [Online]. Tersedia : http://vxheaven.org/vl.php
[27] E. Carrera (2016). erocarrera/pefile. [Online]. Tersedia: https://github.com/erocarrera/pefile.