JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 8
No. 3
Desember
2022

Identifikasi *Malware* Berdasarkan Artefak *Registry Windows 10* Menggunakan *Regshot* dan *Cuckoo*

Yusuf Bambang Setiadji[#1], Dimas Febriyan Priambodo[#2], Muhammad Hasbi[*3], Fadlilah Izzatus Sabila[#4]

[#]*Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara*
Jl. H. Usa , Ciseeng, Bogor 19120

[1]yusuf.setiadji@poltekssn.ac.id
[2]dimas.febriyan@poltekssn.ac.id
[4]fadlilah.izzatus@poltekssn.ac.id

[*]*Informatika, STMIK Sinar Nusantara*
Jl. K.H Samanhudi No.84-86, Purwosari, Laweyan, Surakarta, Jawa Tengah 57149

[3]m.hasbi@sinus.ac.id

Abstrak— ***Malicious software (malware)*** **adalah perangkat lunak yang dibuat dengan tujuan tertentu, seperti mengubah, mencuri, atau merusak data serta mengambil alih sistem.** ***Malware*** **menjalankan tugasnya dengan mengenali faktor-faktor khusus melalui kombinasi parameter dan kondisi pada sistem. Salah satu faktor parameter berjalannya** ***malware*** **adalah sistem operasi. Sebagai sistem operasi dengan pengguna terbanyak, Windows juga memiliki risiko serangan** ***malware*** **tertinggi. Maraknya serangan** ***malware*** **selama 10 tahun terakhir mengharuskan dilakukannya tindakan penanganan insiden** ***malware.*** **Penanganan insiden** ***malware*** **dijalankan bersamaan dengan forensik digital yang digunakan untuk mendapatkan bukti aktivitas** ***malware.*** **Namun, seiring berjalannya waktu** ***malware*** **berkembang dan beradaptasi sehingga menghasilkan jenis-jenis** ***malware*** **dengan kemampuan yang menjadikannya sulit diidentifikasi. Kebutuhan penanganan insiden dapat memanfaatkan artefak digital seperti** ***registry*** **untuk mengidentifikasi keberadaan dan tingkah laku** ***malware*****. Pada penelitian ini dilakukan identifikasi jenis** ***malware*** **berdasarkan artefak** ***registry*** **Windows 10. Penelitian ini melakukan analisis dinamik terhadap 90 sampel** ***malware*** **jenis** ***backdoor, ransomware,*** **dan** ***spyware*** **serta 10 sampel** ***cleanware*** **menggunakan tools Regshot dan Cuckoo yang dijalankan pada lingkungan virtualisasi. Hasil analisis dinamik selanjutnya diekstraksi, direduksi, dihitung, dan dianalisis berdasarkan 34 lokasi** ***registry*** **yang berperan dalam aktivitas** ***malware*** **dan kontaminasi data. Tahapan analisis hasil dilakukan terhadap data analisis dinamik menggunakan Regshot, Cuckoo, dan gabungan kedua tools. Berdasarkan hasil analisis, lokasi dengan modifikasi** ***registry*** **tertinggi pada** ***malware*** **bersifat konsisten sedangkan pada** ***cleanware*** **berubah.** ***Malware*** **jenis** ***backdoor*** **dan ransomware melakukan modifikasi** ***registry*** **tertinggi pada HKLM\SYSTEM, sedangkan spyware melakukan modifikasi** ***registry*** **tertinggi pada HKLM\SOFTWARE\Classes.**

Kata kunci*— *Cuckoo, Malware, Registry Windows, Regshot

I. PENDAHULUAN

Malicious software (malware) adalah perangkat lunak yang didesain untuk tujuan membahayakan. *Malware* memiliki karakteristik tertentu yang dapat diklasifikasikan dalam beberapa jenis yaitu *virus, worm, logic bomb, Trojan horse, backdoor, mobile code*, dan *multiple-threat malware*. Serangan malware mampu digunakan untuk mengganggu operasi komputer, mengumpulkan informasi sensitif, atau mendapatkan akses terhadap komputer. Serangan *malware* selalu mengalami peningkatan dalam kurun waktu 10 tahun terakhir, hingga Oktober 2020 tercatat 1.013.009 serangan *malware* yang terjadi sepanjang tahun [1]. Selama tahun 2019 *malware* menduduki peringkat pertama tren serangan siber dengan persentase 39,3% dalam skala global dan 36% dalam skala nasional [2].

Seiring berjalannya waktu *malware* oleh Adanesi [3] berkembang dan beradaptasi, sehingga dihasilkan berbagai macam jenis *malware* dengan kompleksitas dan karakteristik tingkah lalu berdasarkan sistem yang diserang. *Malware* menjalankan tugasnya dengan mengenali faktor-faktor khusus melalui kombinasi parameter dan kondisi pada sistem. Salah satu faktor yang menjadi parameter berjalannya *malware* adalah sistem operasi [4]. Windows menjadi sistem operasi terbanyak yang digunakan oleh pengguna komputer di dunia dengan persentase 77,12% dan 87,07% untuk pengguna Windows di Indonesia dengan grafik penggunaan yang meningkat setiap tahun [5]. Berbanding lurus dengan peningkatan jumlah pengguna, serangan *malware* yang terjadi pada sistem operasi Windows tercatat mengalami peningkatan pada tahun 2019 dengan total 340.458 serangan pada kategori pengguna bisnis dan personal. *Spyware, backdoor, dan ransomware* termasuk pada daftar 10 *malware* terbanyak yang menyerang kedua kategori tersebut [6].

Maraknya serangan *malware* mendorong dilakukannya usaha penanganan insiden yang disesuaikan dengan karakteristik jenis *malware* [7]. Penanganan insiden dijalankan bersamaan dengan proses forensik digital dalam rangka mengumpulkan data digital guna penyelidikan barang bukti yang diperlukan pada ranah hukum. Kemampuan *malware* yang diprogram dengan tool dan metode tertentu menjadikannya sulit dideteksi [8]. Komponen utama yang dapat digunakan untuk mencari bukti digital dengan mengamati aktivitas *malware* pada Windows adalah *registry*. Windows *registry* memiliki arsitektur yang kompleks namun memberikan banyak manfaat bagi para penyelidik. Forensik *registry* pada *malware* dapat digunakan untuk mengidentifikasi keberadaan dan tingkah laku *malware* [4]. Penelitian yang dilakukan oleh SANS menyatakan bahwa proses pengumpulan dan analisis *malware* membutuhkan waktu dan biaya yang cukup banyak serta membutuhkan teknik yang berbeda-beda. Hal tersebut menjadi suatu tantangan bagi penyelidik dalam melakukan proses forensik digital [8].

Beberapa penelitian telah melakukan studi terkait artefak digital pada sistem operasi Windows, meliputi *registry*, file sistem, dan memori volatile. Penelitian [4] mengamati tingkah laku *malware* pada beberapa jenis sistem operasi Windows untuk menentukan korelasi antara *malware* dengan artefak *registry* sistem operasi. Jenis *malware* yang diteliti meliputi *Trojan, Botnet, dan worm*. Hasil dari penelitian tersebut menunjukkan bahwa artefak Windows *registry* dapat digunakan untuk mengidentifikasi jenis-jenis *malware* pada sistem operasi Windows. Penelitian [9] melakukan analisis malware pada memori volatile. Penelitian tersebut mengombinasikan analisis statik malware dan teknik forensik memori. Penelitian yang dilakukan berhasil mengidentifikasi sampel *malware* dengan persentase kesuksesan sebesar 90%. Pada penelitian [10] diusulkan metode deteksi *malware* jenis *Trojan banking* berdasarkan aktivitas *registry*. Hasil dari penelitian tersebut menunjukkan bahwa seluruh sampel *malware* URSNIF melakukan akses terhadap *registry* HKCU\Software\Microsoft\Windows\CurrentVersion\Run. Penelitian David Orlando [11] juga melakukan analisis menggunakan cuckoo. Cuckoo dinilai mempunyai kemampuan mengisolasi lingkungan yang baik dan dapat digunakan sebagai standar pengujian. Sejalan dengan tersebut Shiva D melengkapinya dengan menggunakan machine learning[12].

Penelitian ini menggunakan Regshot dan Cuckoo melakukan identifikasi 90 sampel *malware* jenis backdoor, spyware, dan ransomware berdasarkan artefak *registry* Windows 10 yang telah dikenali sebagai lokasi *registry* potensial dalam kontaminasi data. Versi Windows yang digunakan adalah Windows 10 yang dirilis pada 29 Juli 2019, merupakan produk terbaru Windows yang paling banyak digunakan dibandingkan versi-versi sebelumnya [13]. Identifikasi dilakukan dengan metode analisis dinamik *malware* menggunakan tool Regshot dan sandbox Cuckoo.

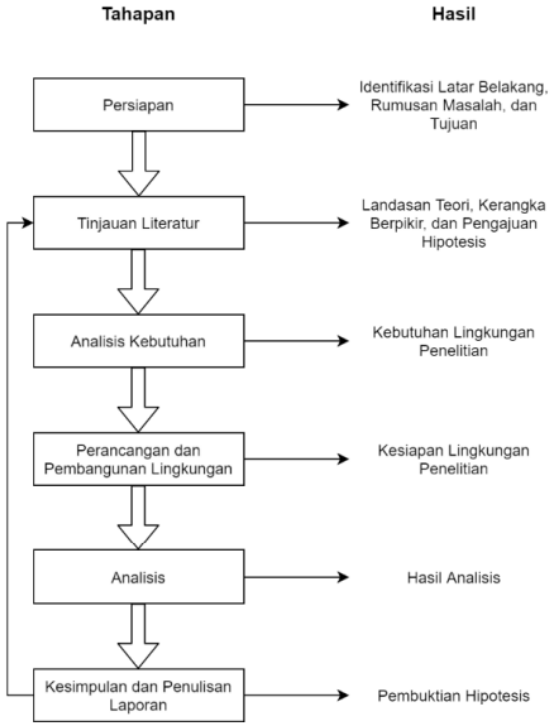
II. PENELITIAN TERKAIT

Penelitian oleh Chathuranga R [9] melakukan analisis terhadap *malware* dengan *hidden behavior* yang disebabkan karena proses enkripsi, *packing*, dan *obfuscating* menggunakan tool Cuckoo dan Volatility. Penelitian Kento K [10] juga Melakukan identifikasi *malware* namun berbeda metode yaitu berdasakan proses eksekusi dan akses *malware* terhadap *registry* pada sistem operasi Windows dengan sandbox Time Freeze 2017, Regshot, dan Process Monitor. Masih dengan pekejaan yangg sama terhadap *malware* oleh Muhammad Ali [4] menambah ragam *malware* seperti *Trojan, Botnet, dan worm*. *Malware* dari berbagai versi sistem operasi Windows digunakan untuk menentukan korelasi dan hubungan antara *malware* dengan artefak *registry*. Penelitian ilhamdi [14] menggunakan analisis dinamis pada windows juga namun terbatas pada trojan dan backdoor dengan 6 sample.

Berbeda dengan penelitian sebelumnya penelitian ini dilakukan menggunakan Regshot dan Cuckoo, penggunaan versi sistem operasi windows 10 sebagai lingkungan eksekusi *malware* dan melakukan analisis *malware* secara dinamik dengan *malware* jenis *backdoor, spyware, dan ransomware*.

III. METODOLOGI PENELITIAN

Penelitian yang dilakukan menerapkan metode kuantitatif melalui rangkaian proses yang ditunjukkan pada Gambar 1.



Gambar. 1 Desain penelitian [15]

A. Persiapan

Pada tahapan ini dilakukan identifikasi masalah, latar belakang, dan tujuan penelitian. Permasalahan diperoleh

melalui studi pendahuluan berdasarkan fakta-fakta empiris. Peningkatan serangan malware dan kemampuannya yang berkembang menjadikan *malware* sulit dideteksi [8]. Mayoritas system informasi yang telah diidentifikasi sebelumnya adalah windows dan *registry* merupakan komponen utama yang dapat diolah lebih lanjut oleh investigator karena *registry* juga mencatat setiap perilaku dari *malware*[4] didukung oleh penelitian dari Kono [10] yang meneliti salah satu jenis malware khususnya *trojan.* Oleh karena itu dalam tahap persiapan telah ditemukan gambaran besar masalah yang dihadapi antara lain bagaimana melakukan investigasi *malware* pada sistem operasi windows.

B. Tinjauan Literatur

Merujuk studi literatur syahputra [16] Rumusan masalah yang telah ditentukan pada tahap pertama menghasilkan hipotesis yang diperoleh dari hasil kajian teoritis terhadap referensi yang relevan dengan masalah. Pada tahapan ini dilakukan tinjauan literatur terhadap teori *malware*, analisis *malware*, *registry* Windows, Regshot, Cuckoo, dan penelitian terkait untuk menentukan kerangka berpikir dan hipotesis penelitian.

Merujuk penelitian sebelumnya Chathuranga Rathnayaka [9] yang Melakukan analisis terhadap *malware* dengan *hidden behaviour* yang disebabkan karena proses enkripsi, packing dan obfuscating menggunakan tool Cuckoo dan Volatility. Penelitian Kento Kono [10] yang melakukan identifikasi *malware* berdasarkan proses eksekusi dan akses *malware* terhadap registry pada sistem operasi Windows dengan sandbox Time Freeze 2017, Regshot, dan Process Monitor. Diperkuat oleh rujukan penelitian oleh Muhammad Ali [4] yang melakukan identifikasi *malware* jenis *Trojan*, *Botnet*, dan *worm* dari berbagai versi sistem operasi Windows untuk menentukan korelasi dan hubungan antara malware dengan artefak *registry*.maka dapat diambil hipotesis yaitu identifikasi jenis *malware* dapat dilakukan dengan mengamati aktivitas artefak *registry* pada sistem operasi Windows 10. Pengujian hipotesis penelitian dilakukan dengan metode eksperimen karena dilakukan pada lingkungan laboratorium dengan perlakukan (*treatment*) tertentu sehingga efek merugikan dari *malware* dapat dibatasi.

C. Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan melalui studi literatur terhadap penelitian sejenis. Tahap analisis kebutuhan dapat disempurnakan setelah proses perancangan penelitian ditentukan sehingga kebutuhan penelitian dapat terpenuhi. Penelitian [10] menggunakan tool Regshot untuk menganalisis *registry* yang terjadi akibat aktivitas *malware*. Penelitian tersebut menggunakan Regshot versi 1.9.0 yang dijalankan pada sistem operasi Windows 10. Pada penelitian [4] diperlukan kebutuhan perangkat keras dan perangkat lunak untuk menjalankan analisis *malware* dengan Cuckoo sandbox. Tabel 1 menunjukkan spesifikasi perangkat keras dan Tabel 2 menunjukkan spesifikasi perangkat lunak yang digunakan pada penelitian serta tabel 3 menerangkan detail spesifikas *virtual machine*.

TABEL I
SPESIFIKASI PERANGKAT KERAS

No	Spesifikasi	Deskripsi
1	Sistem Operasi	Windows 10 Pro
2	Processor	Intel® Core™ i7-4790
3	RAM	16.0 GB
4	Harddisk	1 TB

TABEL II
SPESIFIKASI PERANGKAT LUNAK

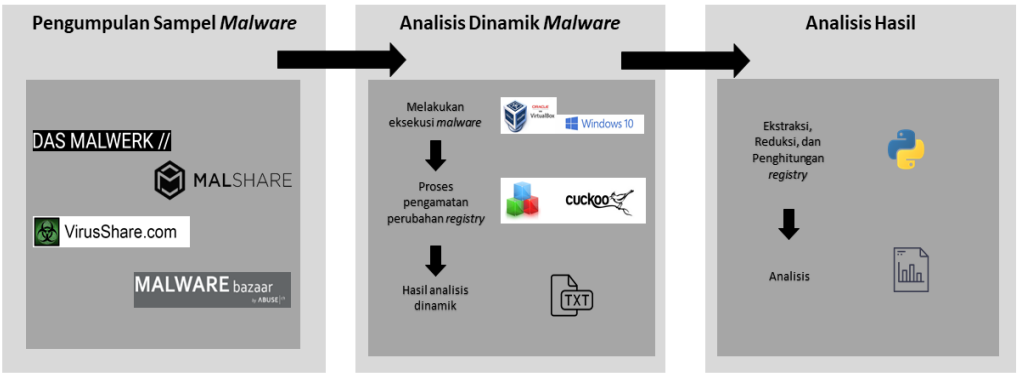
No	Perangkat Lunak	Deskripsi
1	VMWare Workstation Pro 12	Mesin virtual yang dikonfigurasi pada lingkungan bare metal (Windows 10 Pro)
2	Ubuntu 16.04	Sistem operasi yang berperan sebagai Cuckoo host
3	Windows 7, Windows 8.1, dan Windows 10	Sistem operasi yang berperan sebagai Cuckoo guests
4	Cuckoo sandbox	Proses analisis malware
5	Oracle VirtualBox	Mesin virtual yang dikonfigurasi pada lingkungan Ubuntu 16.04

TABEL III
SPESIFIKASI *VIRTUAL MACHINE*

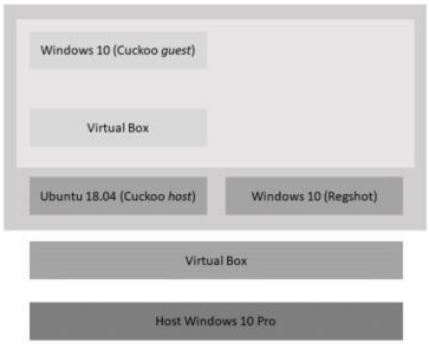
No	Virtual Machine	Deskripsi
1	Mesin Virtual Windows 10	1 CPU RAM 4 GB Hardisk 50 GB Windows 10 versi 1809 Host-only adapter
2	Mesin Virtual Ubuntu 18.04 (Cuckoo Host)	1 CPU enabled nested VT-x/AMD V RAM 10 GB Hardisk 500 GB Ubuntu 18.04 LTS NAT adapter
3	Mesin virtual Windows 10 (Cuckoo Guest)	1 CPU RAM 4 GB Hardisk 50 GB Windows 10 versi 1809 Host-only adapter

D. Perancangan dan Pembangunan Lingkungan

Hasil dari tahapan ini adalah tersusunnya rancangan alur identifikasi *malware*. Perancangan alur penelitian didasarkan pada tujuan penelitian yang telah ditentukan pada tahap-tahap sebelumnya. Alur analisis *malware* yang ditunjukkan pada Gambar 2 terdiri dari tahap pengumpulan sampel *malware*, tahap analisis dinamik *malware*, dan tahap analisis *registry*.



Gambar. 2 Alur identifikasi jenis *malware*



Gambar. 3 Arsitektur lingkungan

Gambar 3 menunjukkan perancangan lingkungan yang digunakan untuk identifikasi *malware*. Laporan dari Cuckoo dapat dilihat melalui website *interface* dalam bentuk laporan dengan format HTML (*Hypertext Markup Language*), PDF (*Portable Document Format*) atau JSON (*JavaScript Object Notation*). Penelitian ini memerlukan informasi aktivitas *registry* yang terdapat pada fitur *behavioral analysis* sejalan dengan penelitian Sagar [17]. Ekstraksi fitur tersebut dapat dilakukan dengan bantuan API Cuckoo yang memanfaatkan hasil analisis dalam format JSON mengikuti langkah dalam penelitain oleh shiva [18]. Fungsi ekstraksi tersebut dijelaskan pada kode program yang ditunjukkan pada Gambar 4

```
procedure EkstraksiRegistry

IMPORT requests, json, os
SET tfile TO open('namafilehasil.txt', 'a')
FOR x IN range(x,y):
   SET REST_URL TO "http://127.0.0.1:8090/tasks/report/"+str(x)
   SET HEADERS TO {"Authorization": "Bearer Q3PI-qK_nZLOlyuLi12TiQ"}
   SET r TO requests.get(REST_URL, headers=HEADERS)
   SET calls TO r.json()["behavior"]["processes"][1]["calls"]
   SET hasil TO []
   SET count TO 0

   FOR i IN calls:
      if(calls[count]["category"]=="registry"):
         IF (calls[count]["api"]=="RegCloseKey"):
            OUTPUT("RegCloseKey")
         ELSE:
            hasil.append(calls[count]["arguments"]["regkey"])
      SET count TO count + 1
   SET hasil TO "," .join(hasil)
   SET hasil TO hasil.lower()

   tfile.write(hasil+"\n")
tfile.close()
```

Gambar. 4 Bagian kode program fungsi pengambilan *registry*

Selanjutnya dilakukan identifikasi jenis *malware* dengan tahapan dijabarkan sebagai berikut:

1. Pengumpulan Sampel *Malware*

Sampel malware pada penelitian ini diperoleh dari beberapa repositori terbuka yang direkomendasikan oleh Lenny Zelster [19]. Penelitian ini menggunakan sampel *malware* yang berasal dari Das Malwerk, Malshare, VirusShare.com, dan *Malware* Bazaar. Proses pengunduhan dan penggunaan sampel *malware* harus dilakukan sesuai dengan syarat dan ketentuan yang ditetapkan oleh penyedia sampel. Pada tahapan ini dilakukan pengunduhan *malware* jenis *ransomware, spyware, dan backdoor* yang digunakan pada tahap selanjutnya.

2. Analisis Dinamik *Malware*

Sampel yang telah diunduh disimpan pada lingkungan virtualisasi untuk dijalankan. Penelitian ini menggunakan lingkungan virtualisasi dengan sistem operasi Windows 10 untuk eksekusi *malware*. Penggunaan tools Regshot dan Cuckoo sandbox digunakan pada proses analisis dinamik *malware* untuk mengamati perubahan *registry* yang terjadi. Pada analisis dinamik *malware* dengan tool Regshot ditetapkan waktu injeksi *malware* selama 5 menit atau ketika *malware* mempengaruhi sistem.

Hasil keluaran dari kedua tools tersebut berupa file dalam format '.txt'. File tersebut berisi informasi aktivitas *registry* yang terdiri dari kunci yang dihapus (keys deleted), kunci yang ditambahkan (*keys added*), nilai yang ditambahkan (*values added*), dan nilai yang dimodifikasi (*values modified*). Hasil analisis tersebut disimpan pada '*Shared Folders*' yang terhubung dengan host.

3. Analisis Aktivitas *Registry*

Keluaran tahapan analisis dinamik *malware* pada Regshot dan Cuckoo digunakan untuk mengamati lokasi *registry* yang berperan pada proses forensik digital berdasarkan potensi kemampuan mengontaminasi data dan aktivitas *malware*. hasil analisis selanjutnya direduksi dengan parameter *registry* yang diinvestigasi dalam proses forensik digital dan dihitung menggunakan *script* pemeriksaan *registry*. Proses reduksi memanfaatkan modul 'line.strip' yang berfungsi untuk mencocokan kata tiap

baris pada file yang dipilih dengan file 'search_list.txt'. Selanjutnya, program menuliskan lokasi *registry* yang menjadi hasil pencocokan kedua file pada file hasil reduksi. Potongan kode dapat dilihat pada gambar 5.

```
procedure ReduksiRegistry

IMPORT sys
SET original_stdout TO sys.stdout
SET list_string TO 'search_list.txt'
SET count TO 0
SET namafile TO INPUT("Masukkan nama file: ")
SET namafiles TO namafile + ".txt"
SET output TO ("hasil_" + namafile)

with open(list_string, 'r') as search_list:
   SET targets TO [line.strip() FOR line IN search_list]
with open(namafiles, 'r') as source_file:
   with open(output, 'w') as f:
      OUTPUT(namafiles)
      FOR line IN source_file:
         IF any(target IN line FOR target IN targets):
            SET sys.stdout TO f
            OUTPUT(line)
            SET sys.stdout TO original_stdout
```

Gambar. 5 Pseudo kode program reduksi *registry*

E. Analisis

Tahapan ini dilakukan implementasi identifikasi *malware* sesuai dengan alur perancangan. Proses analisis dinamik *malware* menghasilkan keluaran berupa kumpulan *registry* yang berperan dalam aktivitas *malware*. Tabel 4 menunjukkan daftar lokasi *registry* yang berperan dalam kontaminasi data dan aktivitas *malware*. Terdapat 34 lokasi *registry* Windows yang telah dikenali sebagai *registry* dengan kemampuan tersebut [4].

TABEL IV
LOKASI *REGISTRY WINDOWS*

No	Lokasi Registry
1	HKLM\SYSTEM\ControlSet001\Control\Nls\CustomLocale\en-US
2	HKLM\SYSTEM\ControlSet001\Control\Nls
3	HKLM\SYSTEM\ControlSet001\Control\Session
4	HKLM\SYSTEM\ControlSet001\Control
5	HKLM\SYSTEM
6	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
7	HKLM\SOFTWARE\Microsoft\ActiveSetup\Installed Components
8	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion
9	HKLM\SOFTWARE\WOW6432Node\Microsoft
10	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer
11	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices
12	HKLM\SOFTWARE\Microsoft\Rpc
13	HKLM\SOFTWARE\Microsoft
14	HKLM\SOFTWARE\Classes\batfile
15	HKLM\SOFTWARE\Classes\exefile
16	HKLM\SOFTWARE\Classes
17	HKLM\SOFTWARE\Policies
18	HKCU\SOFTWARE\Microsoft
19	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folder
20	HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\User Shell
21	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup
22	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
23	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion
24	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer
25	Documents and Settings
26	%systemdrive%\Documents and Settings
27	%Systemdrive%\Users
28	%Systemdrive%\Windows\System32
29	%Systemdrive%\Windows\INF
30	%Systemdrive%\Windows\Globalization\Sorting\sortdefault.nls
31	%Systemdrive%
32	HKCR\Exefile
33	HKCR\Comfile
34	Advanced\Start\ShowDownloads

```
file = open(output, "r")
data = file.read()
Counter = 0
sys.stdout = open("jumlah_registry_" + namafiles, "w")
l0 =
data.count(r"HKLM\SYSTEM\ControlSet001\Control\Nls\CustomLocale\
en-US")
l1 = data.count(r"HKLM\SYSTEM\ControlSet001\Control\Nls")
l2 = data.count(r"HKLM\SYSTEM\ControlSet001\Control\Session")
l3 = data.count(r"HKLM\SYSTEM\ControlSet001\Control")
l4 = data.count(r"HKLM\SYSTEM")
l5 = data.count(r"HKLM\SOFTWARE\Microsoft\Rpc")
l6 =
data.count(r"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\Current
Version")
l7 = data.count(r"HKLM\SOFTWARE\Microsoft")
l8 = data.count(r"HKLM\SOFTWARE\Wow6432Node\Microsoft")
l9 = data.count(r"HKCU\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Windows")
l10 =
data.count(r"HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Setu
p")
l11 =
data.count(r"HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Unin
stall")
```

Gambar. 6 Bagian kode program deklarasi perhitungan *registry*

Gambar 6 menunjukkan program deklarasi penghitungan *registry*. Tahapan ini berjalan setelah proses reduksi pada Gambar 5. Program memanfaatkan *modul* 'data.count' untuk mendeklarasikan lokasi *registry* yang akan dihitung. Terdapat 34 lokasi yang didefinisikan sebagai *array* dengan nama l0 hingga l33 yang merepresentasikan lokasi pada array 0 hingga lokasi pada *array* 33. Penghitungan dilakukan dengan mencocokkan data pada *array* dan file yang dipilih.

```
print(r'HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersi
on:', l6)
if l7 !=0:
    print(r'HKLM\SOFTWARE\Microsoft:', l7)
if l8 !=0:
    print(r'HKLM\SOFTWARE\Wow6432Node\Microsoft:', l8)
if l9 !=0:
    print(r'HKCU\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Windows:', l9)
if l10 !=0:

print(r'HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup:',
l10)
if l11 !=0:

print(r'HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
:', l11)
if l12 !=0:
    print(r'HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion:',
l12)
if l13 !=0:

print(r'HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer:
', l13)
if l14 !=0:
```

Gambar. 7 Bagian kode program pemeriksaan *registry*

Gambar 7 menunjukkan tahapan pemeriksaan *registry* dengan metode pemeriksaan kata sesuai data pada *array*. Tahapan pemeriksaan adalah lanjutan tahapan reduksi *registry* pada Gambar 5. Jika ditemukan data yang sama antara data pada array dengan file yang dipilih, maka program akan mencetak data tersebut dan menyimpannya pada file keluaran hasil reduksi. Proses pemeriksaan dilakukan pada tiap baris file yang dipilih.

```
SET CoList TO data.split("\n")
FOR i IN CoList:
   IF i:
      Counter += 1
Print("Total Registry Ditemukan: ", Counter)
sys.stdout.close()

SET sys.stdout TO original_stdout
```

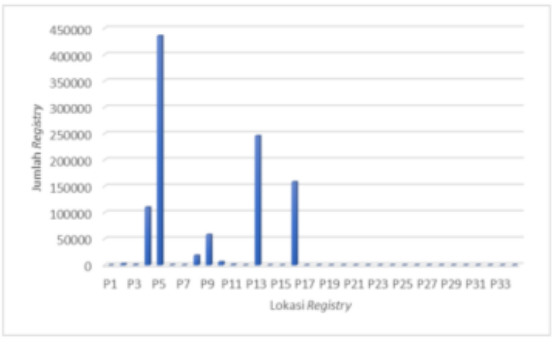
Gambar. 8 Algoritma penghitungan *registry*

Hasil pemeriksaan *registry* pada Gambar 7 dicetak pada file hasil reduksi dengan jarak antar baris lokasi registry. Penghitungan ‘data.count’ pada Gambar 6 digunakan sebagai input dalam program penghitungan registry yang ditunjukkan Gambar 8. Penghitungan dilakukan terhadap hasil reduksi lokasi registry dengan memanfaatkan modul ‘Counter’. Hasil penghitungan registry disimpan dalam file baru dengan struktur penamaan ‘jumlah_registry’ diikuti nama file hasil reduksi yang telah dideklarasikan dalam program pada Gambar 6.

IV. HASIL DAN PEMBAHASAN

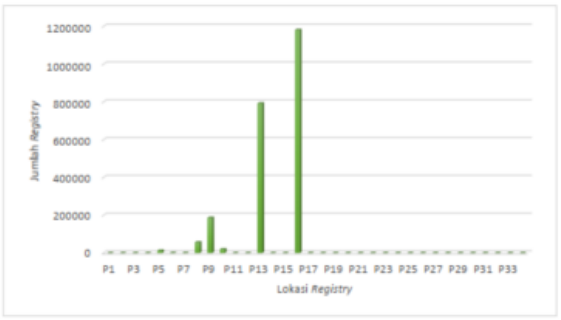
Tahapan analisis dinamik dilakukan pada 90 sampel *malware* dan 10 sampel. Hasil analisis dinamik diolah untuk menentukan lokasi *registry* yang terkena dampak dari aktivitas sampel. Kode program python yang telah dijabarkan sebelumnya digunakan untuk mencatat dan melakukan perhitungan beberapa parameter registry sehingga didapatkan data perubahan registry disertai lokasi disertai jumlahnya.

Gambar 9, 10, 11 dan 12 menunjukkan hasil analisis dinamik *malware* jenis *backdoor*, *ransomware*, *spyware*, *cleanware* menggunakan tool Regshot. Berdasarkan hasil analisis tersebut modifikasi *registry* terjadi pada beberapa lokasi P1-34 sesuai urutan penomoran seperti dalam Tabel 4.

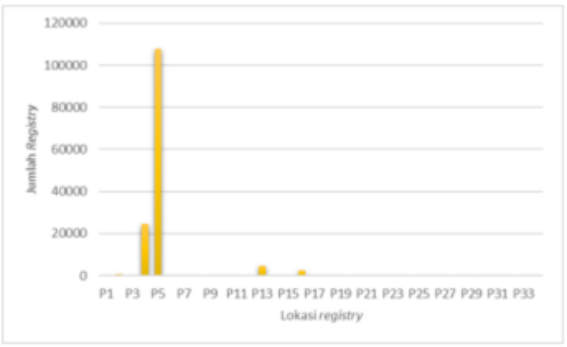


Gambar. 9 Dampak *registry* dari *backdoor* dengan *regshot*

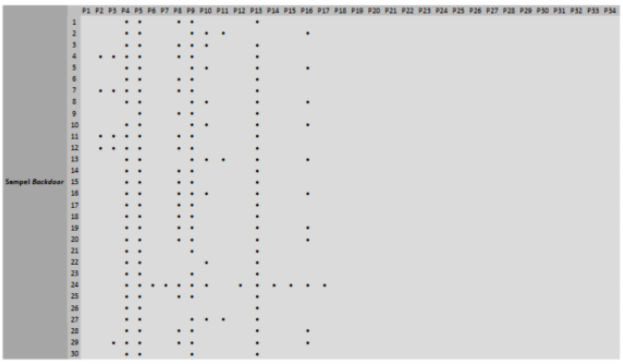
Gambar. 10 Dampak *registry* dari *ransomware* dengan *regshot*



Gambar. 11 Dampak *registry* dari *spyware* dengan *regshot*



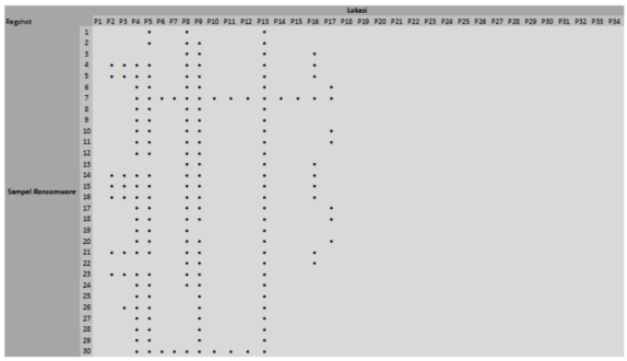
Gambar. 12 Dampak *registry* dari *cleanware* menggunakan *regshot*



Gambar. 13 Pemetaan lokasi *registry* hasil aktivitas *backdoor* menggunakan *regshot*

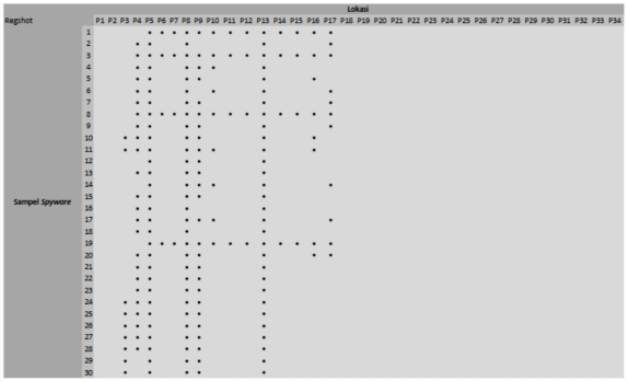
Gambar 13 menunjukkan sebaran sampel *backdoor* dan modifikasinya terhadap *registry* sehinigga dapat dihitung jumlah sampel yang melakukan modifikasi pada setiap *registry*. Terdapat 4 sampel *backdoor* yang memodifikasi *registry* (P2), 4 sampel memodifikasi *registry* (P3), 29

sampel memodifikasi *registry* (P4), 30 sampel memodifikasi *registry* (P5), 1 sampel memodifikasi *registry* (P6), 1 sampel memodifikasi registry (P7), 19 sampel memodifikasi *registry* (P8), 28 sampel memodifikasi *registry* (P9), 10 sampel memodifikasi *registry* (P10), 3 sampel memodifikasi *registry* (P11), 1 sampel memodifikasi *registry* (P12), 27 sampel memodifikasi *registry* (P13), 1 sampel memodifikasi *registry* (P14), 1 sampel memodifikasi *registry* (P15), 12 sampel memodifikasi *registry* (P16), dan 1 sampel memodifikasi *registry* (P17).



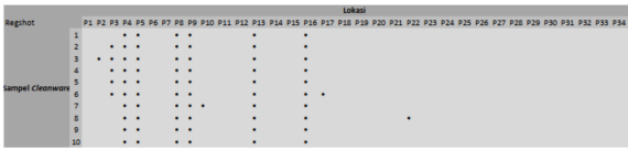
Gambar. 14 Pemetaan lokasi *registry* hasil aktivitas *ransomware* menggunakan *regshot*

Gambar 14 menunjukkan sebaran sampel *ransomware* dan modifikasinya pada setiap *registry*. Terdapat 7 sampel *ransomware* memodifikasi *registry* (P2), 8 sampel memodifikasi registry (P3), 23 sampel memodifikasi *registry* (P4). 27 sampel memodifikasi *registry* (P5), 2 sampel memodifikasi registry (P6), 2 sampel memodifikasi *registry* (P7), 24 sampel memodifikasi *registry* (P8), 28 sampel memodifikasi *registry* (P9), 2 sampel memodifikasi registry (P10), 2 sampel memodifikasi *registry* (P12), 30 sampel memodifikasi registry (P13), 1 sampel memodifikasi *registry* (P14), 1 sampel memodifikasi *registry* (P15), 10 sampel memodifikasi registry (P16), dan 7 sampel memodifikasi *registry* (P17).



Gambar. 15 Pemetaan lokasi *registry* hasil aktivitas *spyware* menggunakan *regshot*

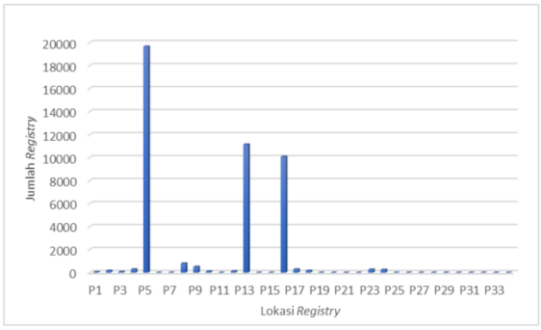
Gambar 15 menunjukkan sebaran data sampel *spyware* dan modifikasi pada setiap *registry*. Terdapat 8 sampel spyware memodifikasi *registry* (P3), 24 sampel memodifikasi *registry* (P4), 30 sampel memodifikasi registry (P5), 4 sampel memodifikasi registry (P6), 4 sampel memodifikasi *registry* (P7), 30 sampel memodifikasi *registry* (P8), 26 sampel memodifikasi *registry* (P9), 10 sampel memodifikasi *registry* (P10), 4 sampel memodifikasi registry (P12), 30 sampel memodifikasi *registry* (P13), 4 sampel memodifikasi *registry* (P14), 4 sampel memodifikasi *registry* (P15), 8 sampel memodifikasi *registry* (P16), dan 11 sampel memodifikasi *registry* (P17).



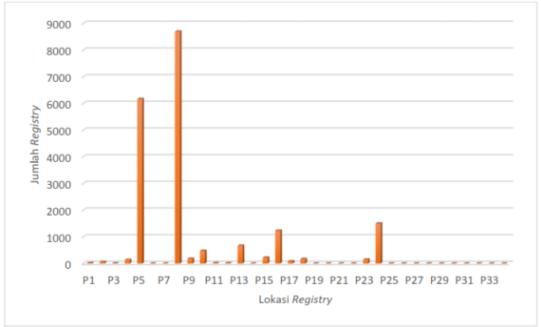
Gambar. 16 Pemetaan lokasi *registry* hasil aktivitas *cleanware* menggunakan *regshot*

Gambar 16 menunjukkan sebaran data sampel *cleanware* dan modifikasinya pada setiap *registry*. Terdapat 1 sampel *cleanware* memodifikasi *registry* (P2), 5 sampel memodifikasi *registry* (P3), 10 sampel memodifikasi *registry* (P4), 10 sampel memodifikasi *registry* (P5), 10 sampel memodifikasi *registry* (P8), 8 sampel memodifikasi *registry* (P9), 1 sampel memodifikasi *registry* (P10), 10 sampel memodifikasi *registry* (P13), 10 sampel memodifikasi *registry* (P16), 1 sampel memodifikasi *registry* (P17), dan 1 sampel memodifikasi *registry* (P22).

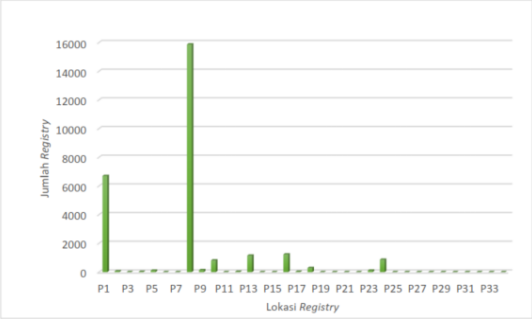
Analisis menggunakan tools yang berbeda yaitu menggunakan Cuckoo didapatkan hasil sebagai berikut. Gambar 17, 18, 19 dan 20 menunjukkan hasil analisis dinamik *malware* jenis *backdoor*, *ransomware*, *spyware*, *cleanware* menggunakan tool Cuckoo.



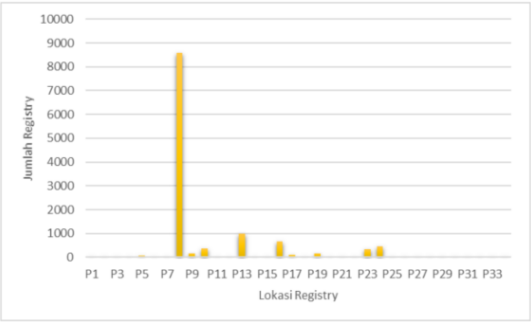
Gambar. 17 Dampak *registry* dari *backdoor* dengan *cukoo*



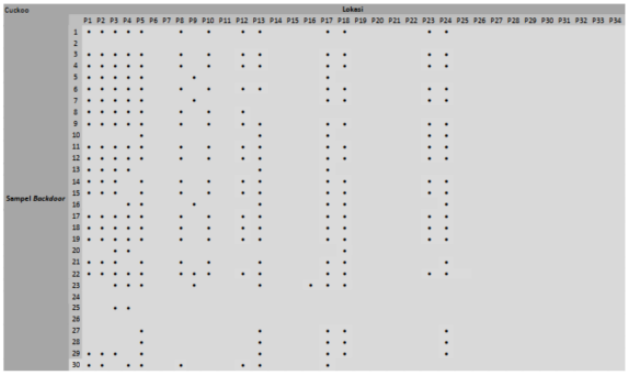
Gambar. 18 Dampak *registry* dari *ransomware* dengan *cukoo*



Gambar. 19 Dampak *registry* dari *spyware* dengan *cukoo*

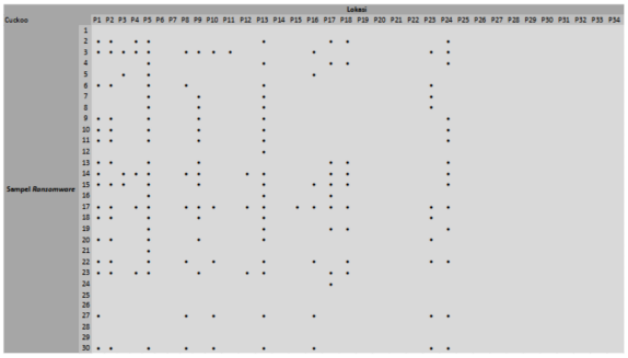


Gambar. 20 Dampak *registry* dari *cleanware* dengan *cukoo*



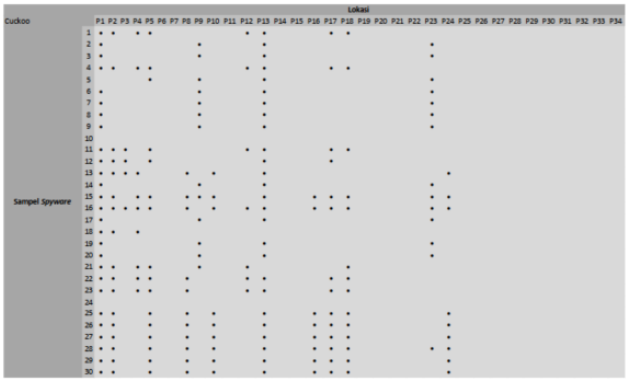
Gambar. 21 Pemetaan lokasi *registry* hasil aktivitas *backdoor* menggunakan *cuckoo*

Gambar 21 menunjukkan bahwa beberapa sampel *backdoor* yang melakukan modifikasi *registry yang* berbeda. Berdasarkan hasil analisis tersebut, dapat dihitung sejumlah 20 sampel backdoor memodifikasi *registry* (P1), 20 sampel memodifikasi *registry* (P2), 23 sampel memodifikasi *registry* (P3), 19 sampel memodifikasi *registry* (P4), 24 sampel memodifikasi *registry* (P5), 16 sampel memodifikasi registry (P8), 5 sampel memodifikasi registry (P9), 15 sampel memodifikasi *registry* (P10), 14 sampel memodifikasi *registry* (P12), 21 sampel memodifikasi *registry* (P13), 1 sampel memodifikasi *registry* (P16), 24 sampel memodifikasi *registry* (P17), 22 sampel memodifikasi *registry* (P18), 15 sampel memodifikasi *registry* (P23), dan 20 sampel memodifikasi *registry* (P24).



Gambar. 22 Pemetaan lokasi *registry* hasil aktivitas *ransomware* menggunakan *cuckoo*

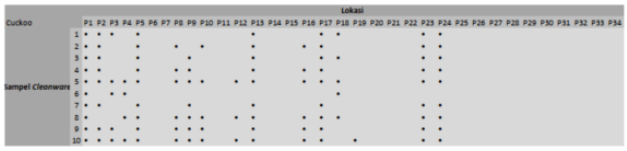
Gambar 22 menunjukkan sebaran sampel *ransomware* yang melakukan modifikasi terhadap registry. Berdasarkan hasil analisis tersebut, sejumlah 17 sampel *ransomware* memodifikasi *registry* (P1), 14 sampel memodifikasi registry (P2), 4 sampel memodifikasi *registry* (P3). 5 sampel memodifikasi *registry* (P4), 21 sampel memodifikasi *registry* (P5), 7 sampel memodifikasi *registry* (P8), 13 sampel memodifikasi *registry* (P9), 6 sampel memodifikasi *registry* (P10), 1 sampel memodifikasi registry (P11), 3 sampel memodifikasi *registry* (P12), 20 sampel memodifikasi *registry* (P13), 1 sampel memodifikasi *registry* HKLM\SOFTWARE\Classes\exefile (P15), 7 sampel memodifikasi registry (P16), 10 sampel memodifikasi *registry* (P17), 9 sampel memodifikasi (P18), 10 sampel memodifikasi *registry* (P23). Dan 14 sampel memodifikasi *registry* (P24).



Gambar. 23 Pemetaan lokasi *registry* hasil aktivitas *spyware* menggunakan *cuckoo*

Gambar 23 menunjukkan sebaran sampel *spyware* yang melakukan modifikasinya terhadap *registry*. Berdasarkan hasil analisis tersebut, terdapat 27 sampel spyware memodifikasi *registry* (P1), 17 sampel memodifikasi *registry* (P2), 4 sampel memodifikasi *registry* (P3), 9 sampel memodifikasi registry (P4), 16 sampel memodifikasi *registry* (P5), 11 sampel memodifikasi *registry* (P8), 13 sampel memodifikasi *registry* (P9), 8 sampel memodifikasi *registry* (P10), 7 sampel memodifikasi *registry* (P12), 26 sampel memodifikasi *registry* (P13), 8 sampel memodifikasi *registry* (P16), 14 sampel memodifikasi *registry* (P17), 14 sampel

memodifikasi *registry* (P18), 14 sampel memodifikasi *registry* (P23), dan 9 sampel memodifikasi *registry* (P24).



Gambar. 24 Pemetaan lokasi *registry* hasil aktivitas *cleanware* menggunakan *cuckoo*

Gambar 24 menunjukkan sebaran data sampel *cleanware* yang melakukan modifikasi terhadap *registry*. Terdapat 8 sampel *cleanware* memodifikasi *registry* (P1), 8 sampel memodifikasi registry (P2), 5 sampel memodifikasi *registry* (P3), 4 sampel memodifikasi registry (P4), 9 sampel memodifikasi *registry* (P5), 6 sampel memodifikasi *registry* (P8), 7 sampel memodifikasi registry (P9), 5 sampel memodifikasi *registry* (P10), 3 sampel memodifikasi *registry* (P12), 9 sampel memodifikasi *registry* (P13), 6 sampel memodifikasi *registry* (P16), 9 sampel memodifikasi *registry* (P17), 5 sampel memodifikasi *registry* (P18), 1 sampel memodifikasi *registry* (P19). 9 sampel memodifikasi registry (P23), dan 9 sampel memodifikasi *registry* (P24).

Hasil analisis dari kedua tool regshoot dan cockoo dikompilasi untuk dapat diambil kesimpulan lebih lengkap. Sehingga dapat diambil kesimpulan bahwa jenis malware tertentu mempunyai kecenderungan mengubah registry pada bagian apa dan dimana lokasinya. Rangkuman dari perbandingan analisis *malware* dengan regshoot dan cuckoo dilampirkan dalam tabel 5.

TABEL V
HASIL PERBANDINGAN ANALISIS DINAMIK *MALWARE*

Sampel	Paramater	Tool		
		Regshot	**Cuckoo**	**Gabungan**
backdoor	Jumlah Lokasi *Registry*	16	15	19
	Lokasi *Registry* Terbanyak	HKLM\SYSTEM	HKLM\SYSTEM dan HKLM\SOFTWARE\Policies	HKLM\SYSTEM
	Jumlah Modifikasi *Registry*	1.034.670	28.075	
	Lokasi dengan Jumlah Modifikasi Tertinggi	HKLM\SYSTEM	HKLM\SYSTEM	HKLM\SYSTEM
ransonware	Jumlah Lokasi *Registry*	15	17	19
	Lokasi *Registry* Terbanyak	HKLM\SOFTWARE\Microsoft	HKLM\SYSTEM	HKLM\SOFTWARE\ Microsoft
	Jumlah Modifikasi *Registry*	1.942.401	19.801	
	Lokasi dengan Jumlah Modifikasi Tertinggi	HKLM\SYSTEM	HKLM\SYSTEM	HKLM\SYSTEM
Spyware	Jumlah Lokasi *Registry*	14	15	18
	Lokasi *Registry* Terbanyak	HKLM\SYSTEM, HKLM\SOFTWARE\WOW6432 Node\Microsoft\Windows\Curren tVersion, dan HKLM\SOFTWARE\Microsoft	HKLM\SYSTEM\ControlSet001\C ontrol\Nls\CustomLocale\en-US	HKLM\SOFTWARE\ Microsoft
	Jumlah Modifikasi *Registry*	2.250.995	27.458	
	Lokasi dengan Jumlah Modifikasi Tertinggi	HKLM\SOFTWARE\Classes	HKLM\SOFTWARE\Classes	HKLM\SOFTWARE\ Classes
Cleanware	Jumlah Lokasi *Registry*	11	16	17
	Lokasi *Registry* Terbanyak	HKLM\SYSTEM\ControlSet001\ Control, HKLM\SYSTEM, HKLM\SOFTWARE\WOW6432 Node\Microsoft\Windows\Curren tVersion, HKLM\SOFTWARE\Microsoft, dan HKLM\SOFTWARE\Classes	HKLM\SYSTEM, HKLM\SOFTWARE\Microsoft, HKLM\SOFTWARE\Policies, HKCU\SOFTWARE\Microsoft\Wi ndows\CurrentVersion, dan HKCU\SOFTWARE\Microsoft\Wi ndows\CurrentVersion\Explorer	HKLM\SYSTEM dan HKLM\SOFTWARE\ Microsoft
	Jumlah Modifikasi *Registry*	140.668	11.997	
	Lokasi dengan Jumlah Modifikasi Tertinggi	HKLM\SYSTEM	HKLM\SOFTWARE\WOW6432N ode\Microsoft\Windows\CurrentVe rsion	HKLM\SYSTEM

V. KESIMPULAN

Artefak *registry* windows 10 dapat digunakan untuk mengidentifikasi *malware*. Dari penelitian ini juga dihasilkan 4 parameter yang dapat digunakan untuk menentukan jenis *malware* yaitu jumlah lokasi *registry*, lokasi *registry* terbanyak, jumlah modifikasi dan lokasi jumlah modifikasi tertinggi. Berdasarkan hasil analisis, lokasi dengan modifikasi *registry* tertinggi ditemukan pada *backdoor*, *spyware* dan *ransomware* bersifat konsisten sedangkan pada *cleanware* berubah. *Malware* jenis *backdoor* dan ransomware melakukan modifikasi *registry* tertinggi pada HKLM\SYSTEM, sedangkan spyware melakukan modifikasi *registry* tertinggi pada HKLM\SOFTWARE\Classes. Sehingga dengan memanfaatkan parameter ini jika lokasi registry terbanyak dan banyak terjadi perubahan pada HKLM/SYSTEM maka dapat dikategorisasikan sebagai backdoor. Namun jika lokasi registry terbanyak pada HKLM\SOFTWARE\Microsoft dan modifikasi terbanyak

ditemukan pada HKLM\SYSTEM dapat dikategorisasikan sebagai malware jenis ransomware. Kategorisasi terakhir untuk spyware dapat dilihat dari lokasi modifikasi registry pada HKLM\SOFTWARE\Classes.

REFERENSI

[1] "Malware Statistics & Trends Report | AV-TEST - Penelusuran Google." https://www.google.com/search?q=Malware+Statistics+%26+Trends+Report+%7C+AV-TEST&oq=Malware+Statistics+%26+Trends+Report+%7C+AV-TEST&aqs=chrome.0.69i59.1172j0j4&sourceid=chrome&ie=UTF-8 (accessed Dec. 17, 2020).

[2] PUSOPSKAMSINAS, "Indonesia Cyber Security Monitoring Report 2019," *Indones. Secur. Incid. Response Team Internet Infrastruct.*, p. 42, 2020, [Online]. Available: https://cloud.bssn.go.id/s/nM3mDzCkgycRx4S/download.

[3] R. Adenansi and L. A. Novarina, "Malware dynamic," *J. Educ. Inf. Commun. Technol.*, vol. 1, no. 1, pp. 37–43, 2017.

[4] M. Ali, S. Shiaeles, N. Clarke, and D. Kontogeorgis, "A proactive malicious software identification approach for digital forensic examiners," *J. Inf. Secur. Appl.*, vol. 47, pp. 139–155, 2019, doi: https://doi.org/10.1016/j.jisa.2019.04.013.

[5] "Desktop Operating System Market Share Worldwide | Statcounter Global Stats." https://gs.statcounter.com/os-market-share/desktop (accessed Dec. 17, 2020).

[6] M. Labs, "2020 State of Malware Report," *Malwarebytes*, 2020. https://resources.malwarebytes.com/files/2020/02/2020_State-of-MalwareReport.pdf. (accessed Dec. 17, 2020).

[7] A. Verma, M. S. Rao, A. K. Gupta, W. Jeberson, and V. Singh, "A LITERATURE REVIEW ON MALWARE AND ITS ANALYSIS," *Int J Cur Res Rev*, vol. 5, Jan. 2013.

[8] S. Almarri and P. Sant, "Optimised Malware Detection in Digital Forensics," *Int. J. Netw. Secur. Its Appl.*, vol. 6, pp. 1–15, Jan. 2014, doi: 10.5121/ijnsa.2014.6101.

[9] C. Rathnayaka and A. Jamdagni, "An Efficient Approach for Advanced Malware Analysis Using Memory Forensic Technique," in *2017 IEEE Trustcom/BigDataSE/ICESS*, 2017, pp. 1145–1150, doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.365.

[10] K. Kono, S. Phomkeona, and K. Okamura, "An Unknown Malware Detection Using Execution Registry Access," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018, vol. 02, pp. 487–491, doi: 10.1109/COMPSAC.2018.10281.

[11] D. Pelaez, D. Díaz-López, D. Sepúlveda-Alzate, and D. Cabuya-Padilla, "Building malware classificators usable by State security agencies," *ITECKNE*, vol. 15, pp. 107–121, Dec. 2018, doi: 10.15332/iteckne.v15i2.2072.

[12] S. D. Sl, A. M.A, and C. Jaidhar, *Windows malware detection based on cuckoo sandbox generated report using machine learning algorithm*. 2016.

[13] "Lifecycle FAQ - Windows | Microsoft Docs." https://docs.microsoft.com/en-us/lifecycle/faq/windows (accessed Dec. 17, 2020).

[14] Y. Ilhamdi and Y. N. Kunang, "Analisis Malware Pada Sistem Operasi Windows Menggunakan Teknik Forensik," *Bina Darma Conf. Comput. Sci.*, pp. 256–264.

[15] P. D. Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*, 3rd ed. Bandung: CV Alfabeta, 2021.

[16] R. Syahputra and Syaifudin, "Studi Literatur Analisis Malware Menggunakan Metode Analisis Dinamis dan Statis," *J. Jar. Komput. dan Keamanan*, vol. 1, no. 1, pp. 14–24, 2020.

[17] M. A. A. m. Mr. Sagar . Sh, "A Review Paper on Effective Behavioral Based Malware Detection and Prevention Techniques for Android Platform," vol. 10, no. 1, pp. 901–907, 2017.

[18] S. D. Sl and C. Jaidhar, "Windows malware detection system based on LSVC recommended hybrid features," *J. Comput. Virol. Hacking Tech.*, vol. 15, Jun. 2019, doi: 10.1007/s11416-018-0327-9.

[19] "Free Malware Sample Sources for Researchers." https://zeltser.com/malware-sample-sources/ (accessed Dec. 17, 2020).