Metode Klasifikasi Malware Berdasarkan *Pattern Assembly* Suatu Malware

Anggrahito
Badan Siber dan Sandi Negara
Email: anggrahito@bssn.go.id

Ramadhan Ibrahim
Badan Siber dan Sandi Negara
Email:
ramadhan.ibrahim@bssn.go.id

Tomy Gunawan
Badan Siber dan Sandi Negara
Email:
Tomy.Gunawan@bssn.go.id

Abstrak—**Seperti yang sudah kita ketahui bahwa perkembangan teknologi siber saat ini berkembang pesat. Pada penelitian ini membahas terkait malware. Penelitian difokuskan pada metode dalam klasifikasi malware. Permasalahan saat ini yang dihadapi adalah ketika terdapat malware baru yang menginfeksi suatu sistem, infrastruktur, dan jaringan adalah sulitnya dilakukan klasifikasi jenis malware. Adanya metode ini dapat mempersingkat waktu dan memudahkan *Computer Emergency Response Team* (CERT) untuk melakukan investigasi.**

Kata kunci; malware, reverse engineering, bash, python.

Abstract— **As we already know that the development of cyber technology is currently growing rapidly. In this research, it is related to malware. Research is focused on methods in classifying malware. The current problem faced is when there is new malware that infects a system, infrastructure, and network, it is difficult to classify types of malware. The existence of this method can shorten the time and facilitate the Computer Emergency Response Team (CERT) to conduct investigations.**

Keywords; malware, reverse engineering, bash, python.

I. PENDAHULUAN

A. Latar Belakang

Infeksi malware adalah ancaman yang paling sering ditemui dalam keamanan komputer. Menurut "Laporan Landscape Ancaman ENISA 2017" beberapa vendor antivirus mendeteksi lebih dari 4 juta sampel malware per hari dan lebih dari 700 juta sampel pada Q1 2017 saja [1]. Dengan adanya metode penggklasifikasian malware dapat mempermudah dan mempercepat CERT untuk memfokuskan teknik investigasinya terhadap suatu malware. Selain itu, Membedakan dan mengklasifikasikan berbagai jenis malware dari satu sama lain adalah penting untuk lebih memahami bagaimana mereka dapat menginfeksi komputer dan perangkat, tingkat ancaman yang mereka hadapi dan bagaimana melindungi mereka [2].

Untuk menjawab permasalahan diatas, maka dibutuhkan metode cepat dalam klasifikasi malware. Pada penelitian ini, implementasi dilakukan dengan memanfaatkan beberapa tools berbasis Open Source yaitu radare2, python, dan bash. Oleh karena itu, dengan diimplementasikannya metode ini dapat mempercepat proses klasifikasi malware ketika ditemukan malware jenis baru.

B. Ruang Lingkup

Pada penelitian ini, akan melakukan implementasi dan pengujian dalam bentuk *Proof of Concept* dalam skala lab. Pengujian dilakukan berdasarkan 15 sampel malware untuk dilakukan ekstraksi *pattern assembly* dan 3 sample malware untuk pengujian klasifikasi malware. Semua sample malware yang digunakan berupa *executable* atau binaries.

C. Metode Penelitian

Untuk melakukan pembuktian dan pengujian, penelitian ini menggunakan metode eksperimen. Pada TABEL I menunjukkan tahapan yang dilakukan pada penelitian ini [3].

TABEL I. TABEL PENGETAHUAN DAN SAINS

Aspek	Tahap 1	Tahap 2	Tahap 3
A. Pengetahuan	1. Keyakinan	2. Justifikasi	3. Kebenaran
	↓	↓	↓
B. Sains	1. Teori	2. Eksperimen	3. Kebenaran sains
	↓		
	Pemodelan		
	Engineering		

Bedasarkan TABEL I, terdapat tiga tahapan yang harus dipenuhi berdasarkan aspek pengetahuan dan sains. Pada penelitian ini pemodelan dilakukan menggunakan flowchart.

II. LANDASAN TEORI

A. Malware

Definisi malware atau *malicious software* (perangkat lunak berbahaya) adalah program kode yang digunakan penyerang untuk merusak atau menyalahgunakan sistem [4]. Terdapat berbagai macam jenis malware, diantaranya Infectors, Network worms, Trojan horses, Backdoors, Remote-access Trojans, Information stealers, Ransomware, Scareware, Fakeware, Greyware. Pada penelitian berfokus pada 4 jenis malware yaitu: empat ransomware, tiga trojan, tiga virus, dan dua worm.

B. Reverse Engineering

Definisi *Reverse Engineering* (rekayasa balik) adalah proses di mana objek buatan manusia didekonstruksi untuk mengungkapkan desain, arsitektur, atau untuk mengekstraksi pengetahuan dari objek [5]. Pada penelitian

ini, proses *reverse engineering* dilakukan menggunakan tools radare2. Dipilihnya tools radare2 pada penelitian ini karena memiliki kemampuan untuk melakukan *disassemble* suatu *executeable* maupun binaries untuk kemudian memunculkan informasi *pattern assembly*. Adanya adanya *pattern assembly* tersebut dapat digunakan untuk klasifikasi malware ketika didapati malware yang belum teridentifikasi.

C. Python

Python merupakan Bahasa pemrograman yang terlihat mudah namun memiliki kemampuan yang stabil dan tidak memakan memori yang cukup besar ketika dijalankan. Dengan menggunakan Bahasa pemrograman python dapat mempermudah seorang programmer untuk mengatasi masalah yang kompleks [6]. Pengguaan python pada penelitian ini karena memiliki library radare2 sehingga mendukung untuk melakukan parsing hasil ekstraksi *pattern assembly* pada *executable* atau binaries malware dan pencocokan antara hasil ekstraksi *pattern assembly* dengan output skor. Selain digunakan untuk melakukan parsing hasil ekstraksi, pada penelitian ini python juga diguanakn untuk memvisualisasikan output hasil proses klasifikasi menggunakan modul poltly pada python.

D. Bash

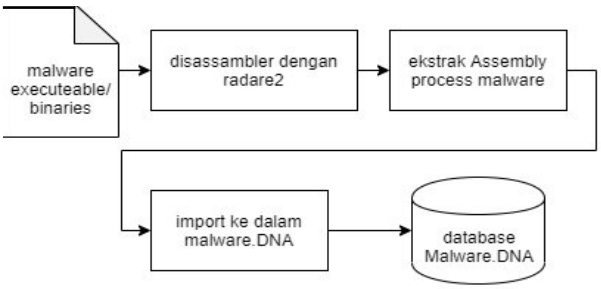
Shell Bourne-Again lebih umum disebut bash adalah salah satu program di Proyek GNU yang merupakan shell paling banyak digunakan di lingkungan * NIX. *Bourne-Again shell* adalah pengembangan dari Bourne shell. Nama tersebut diambil dari penciptanya, yaitu Stephen Bourne. bash sendiri kemudian dikembangkan oleh Brian Fox [7]. Pengguaan bash pada penelitian ini adalah untuk melakukan regex processing dan otomatisasi serangkaian command linux.

III. IMPLEMENTASI

A. Pemodelan

Pada penelitian ini pemodelan menggunakan flowchart agar mudah dipahami. Terdapat dua proses yang dijalankan ketika melakukan klasifikasi malware. Berikut ini adalah penjelasan dari masing masing proses yang akan diimplementasikan:

1) Proses Ekstraksi

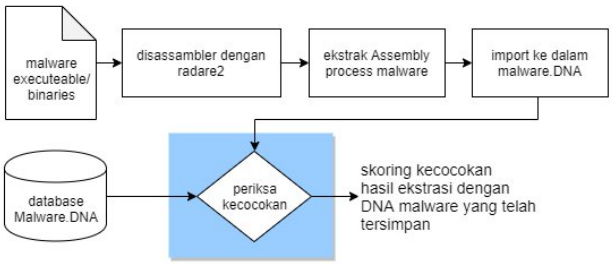


Gambar 1. flowchart metode ekstraksi *pattern assembly* malware

berdasarkan Gambar 1, tahapan yang dilakukan adalah sebagai berikut:

1. mendapatkan/ memiliki malware dalam bentuk executeable ataupun binaries.
2. Melakukan proses disassamble terhadap malware tersebut sehingga dapat diketahui *pattern assembly* yang berjalan pada malware.
3. Melakukan ekstraksi *pattern assembly* ke dalam bentuk file txt.
4. Melakukan parsing *pattern assembly* pada file dengan format .DNA.
5. Menyimpan file .DNA pada satu file direktori.

2) Proses Klasifikasi



Gambar 2. flowchart metode klasifikasi malware

berdasarkan Gambar 2, tahapan yang dilakukan adalah sebagai berikut:

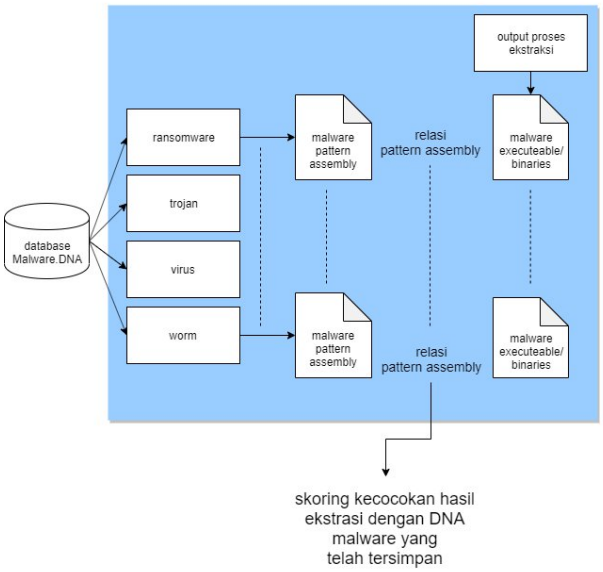
1. mendapatkan/ memiliki malware dalam bentuk executeable ataupun binaries.
2. Melakukan proses disassamble terhadap malware tersebut sehingga dapat diketahui *pattern assembly* yang berjalan pada malware.
3. Melakukan ekstraksi *pattern assembly* ke dalam bentuk file txt.
4. Melakukan parsing *pattern assembly* pada file dengan format .DNA.
5. Ouput dari hasil parsing *pattern assembly* kemudian dicocokkan dengan setiap *pattern assembly* yang telah teridentifikasi dan disimpan.
6. Hasil pemeriksaan kecocokan hasil ekstraksi dengan *pattern assembly* yang telah tersimpan berupa skor kecocokan.

3) Proses penilaian

Penilaian dilakukan berdasarkan banyaknya pattern yang ber relasi antara hasil ekstraksi dengan yang sudah ada di dalam database. Adapun pengertian relasi yaitu, kumpulan pasangan berurut yang mengandung satu objek dari setiap set. Jika objek x berasal dari himpunan pertama dan objek y dari himpunan kedua, maka objek dikatakan terkait jika pasangan berurutan (x, y) berada dalam relasi [8].

Dengan pendekatan relasi tersebut, maka penlilaian dilakukan jika objek *pattern assembly* yang berasal dari hasil ekstraksi dan objek *pattern assembly* dari database, maka objek dikatakan terkait jika pasangan berurutan (x, y) berada dalam relasi. Adapun relasi yang dimaksud adalah memiliki pattern yang sama.

Berdasarkan pendekatan tersebut, flowchart proses penilaian dilakukan seperti pada gambar 3 dibawah ini:



Gambar 3. flowchart proses penilaian

proses penilaian terbagi atas dua yaitu:

- proses penilaian sebagaian
  proses penilaian sebagian dilakukan dengan cara, output proses ekstraksi sebuah malware dikatakan berelasi dengan file *pattern assembly* yang terdapat pada masing-masing jenis malware yang sudah terdaftar dan masuk ke dalam database jika terdapat sebagian *pattern assembly* yang sama. Output yang ditampilkan berupa skor, tergantung dari banyaknya *pattern assembly* yang sama.
- Proses penilaian identik
  proses penilaian identik dilakukan dengan cara, output proses ekstraksi sebuah malware dikatakan berelasi dengan file *pattern assembly* yang terdapat pada masing-masing jenis malware yang sudah terdaftar dan masuk ke dalam database jika seluruh *pattern assembly* memiliki nilai yang sama. Output yang ditampilkan berupa statement, jika benar identik maka akan ditampilkan statement "malware ini identik".

B. Engineering

Pada bagian ini akan menjelaskan penggunaan script yang sudah dikembangkan untuk memudahkan dalam ekstraksi *pattern assembly* suatu malware. Proses ekstraksi *pattern assembly* dilakukan dengan menjalankan scirpt bash menggunakan command `bash script3.sh`. berikut ini adalah output yang ditampilkan ketikan menjalankan script3.sh:

```
=================================================================

total 15072
-rwxrwxrwx        ...        cerber.exe
-rw-rw-r--        ...        cryptowall.bin
-rw-rw-r--        ...        jigsaw
     .
     .
     .
-rw-rw-r--        ...        radamant.exe

-rw-rw-r--        ...        satana.exe
-rw-rw-r--        ...        wannacry.exe
=================================================================

masukkan nama executebale :
wannacry.exe

masukkan nama output DNA malware :
wannacry

memulai proses ekstraksi..
```

Berdasarkan contoh proses di atas, percobaan ini menggunakan ransomware wannacry. Executetable yang digunakan adalah wannacry.exe dan output nama *pattern assembly* yang diinginkan adalah wannacry. Berdasarkan input yang dimasukkan pada script.sh, output dari hasil proses tersebut adalah wannacry.DNA. Berikut ini adalah isi dari wannacry.DNA:

```
push ebp
mov ebp, esp
in al, dx
push -1
ljmp [eax - 0x78]
push 0x40d488
mov ah, dl
aam 0x40
inc eax
add byte [eax - 0xc], ch
.
.
.
mov dword [ebp - 0x30], ebx
mov dword [ebp - 0x30], ebx
pop ebp
```

berdasarkan output yang telah dihasilkan, pada wannacry.DNA terdapat 254 line patern yang akan dijadikan parameter untuk dicocokan dengan database malware yang ada.

Kemudian untuk memeriksa kecocokan suatu *pattern assembly* malware antara file .DNA dilakukan menggunakan script python `persen.py`. berikut ini adalah contoh penggunaan script python untuk membandingkan file wannacry.DNA dengan wannacry.DNA tanpa ada perubahan pada file tersebut.

```
Input:
python persen.py wannacry.DNA wannacry.DNA

Output:
100.0%
```

Berdasarkan output yang ditampilkan, jika nilai yang didapat adalah 100.0 % maka dapat dikatakan bahwa *pattern assembly* yang diujikan dengan yang sudah ada pada database adalah identik.

Lalu untuk memastikan bahwa jika terdapat perbedaan antara file dan script mampu mendeteksi perbedaanya dilakukan sedikit perubahan pada file wannacry.DNA

dengan menhapus line pertama dan menimpannya dengan nama wannacry_modif.DNA. berikut ini adalah output yang ditampilkan ketika script python dijalankan:

```
Input:
python persen.py wannacry.DNA
wannacry_modif.DNA


Output:
99.8943785941%
```

C. Justifikasi

Pada bagian ini akan menjelaskan hipotesis penelitian. Adapun hipotesis yang muncul pada penelitian ini yaitu:

- *pattern assembly* suatu malware dapat dimanfaatkan untuk proses klasifikasi malware.
- Beberapa malware pasti memiliki kemiripan *pattern assembly* antara satu dengan yang lainnya walaupun dengan skor kecil.

D. Eksperimen

Pada penelitian ini Eksperimen terdapat dua proses yang dilakukan, yaitu: pertama, mengekstraksi semua sampel malware yang telah dimiliki kemudian dimasukkan kedalam tiga folder. Folder tersebut adalah ransomware, trojan, virus. Kedua, melakukan ujicoba klasifikasi dengan mengguanakn 3 sample malware yang belum diklasifikasi. Berikut ini adalah penjelasan dari dua proses yang dilakukan:

1) Ekstraksi pattern assembly

Terdapat 14 sample malware yang diekstraksi, berikut adalah list malware yang diekstraksi:

TABEL II. TABEL EKSTRAKSI *PATTERN ASSEMBLY*

No	Nama malware	Nama output pattern assembly
1	cryptolocker.exe	cryptolocker.DNA
2	cryptowall.bin	cryptowall.DNA
3	Matsnu.exe	matsnu.DNA
4	satana.exe	satana.DNA
5	wannacry.exe	wannacry.DNA
6	Artemis.exe	Artemis.DNA
7	bechiro.exe	bechiro.DNA
8	DarkTequila.exe	DarkTequila.DNA
9	Kovter.exe	Kovter.DNA
10	NSIS.exe	NSIS.DNA
11	Hybris.exe	Hybris.DNA
12	klez.exe	klez.DNA
13	slammer.exe	slammer.DNA
14	stuxnet.exe	stuxnet.DNA
15	Swen.exe	Swen.DNA

Kemudian masing-masing hasil ekstraksi *pattern assembly* dimasukkan ke dalam folder berdasarkan jenis malware. Berikut ini adalah tabel pemetaan ekstraksi *pattern assembly* ke dalam folder berdasarkan jenis malware:

TABEL III. PEMETAAN EKSTRAKSI *PATTERN ASSEMBLY* KE DALAM FOLDER

Nama folder	Nama output pattern assembly
Ransomware	cryptolocker.DNA
	cryptowall.DNA
	matsnu.DNA
	satana.DNA
	wannacry.DNA
Trojan	Artemis.DNA
	bechiro.DNA
	DarkTequila.DNA
	Kovter.DNA
	NSIS.DNA
Virus / Worm	Hybris.DNA
	klez.DNA
	slammer.DNA
	stuxnet.DNA
	Swen.DNA

2) Klasifikasi

Pada percobaan klasifikasi, terdapat dua proses pengujian yang akan dilakukan yaitu: proses penilaian sebagian dan proses penilaian identik. Berikut ini adalah penjelasan dari masing-masing proses penilaian:

- Proses penilaian sebagian:

Pada proses ini digunakan sample malware yang belum teridentifikasi sehingga program akan memeriksa relasi antara setiap *pattern assembly* hasil ekstraksi dengan *pattern assembly* malware yang sudah ada pada database. Adapun malware yang akan dijadikan sampel adalah :

- LMclicker.exe
- NIMDA.exe
- petya.bin

Berikut ini adalah salah satu contoh proses klasifikasi malware dengan mengambil sample LMclicker.exe, command yang dijalankan adalah:

```
bash class.sh
```

output yang ditampilkan class.sh dijalankan untuk klasifikasi malware adalah sebagai berikut:

```
masukkan nama executebale :
LMclicker.exe

masukkan nama output DNA malware :
LMclicker

memulai proses pengklasifikasian...

=============================================
          ransomware
---------------------------------------------
ransomware/cryptolocker.DNA
18
ransomware/cryptowall.DNA
17
ransomware/matsnu.DNA
10
ransomware/satana.DNA
15
ransomware/wannacry.DNA
16
---------------------------------------------
skoring berdasarkan DNA ransomware:
15
=============================================


=============================================
          trojan
---------------------------------------------
trojan/Artemis.DNA
18
trojan/bechiro.DNA
15
trojan/DarkTequila.DNA
17
trojan/Kovter.DNA
22
trojan/NSIS.DNA
15
---------------------------------------------
skoring berdasarkan DNA trojan:
17
=============================================


=============================================
          virus / worm
---------------------------------------------
virus/Hybris.DNA
19
virus/klez.DNA
15
virus/slammer.DNA
10
virus/stuxnet.DNA
18
virus/Swen.DNA
15
---------------------------------------------
skoring berdasarkan DNA virus/worm:
15
=============================================

++++++++++++++++++++++++++++++++++++++++++++

skor penilaian
---------------
ransomware : 15
trojan     : 17
virus/wrom : 15

kesimpulan :
lebih dominan trojan
```

Total skoring :

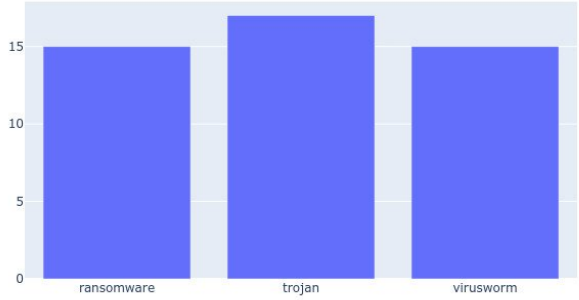
$$TS = \sum PM / \sum PM$$

Keterangan :

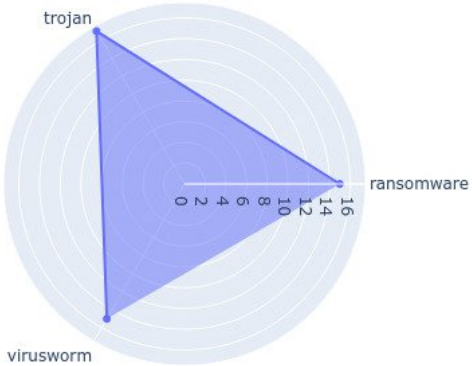
- TS : Total Skoring
- $\sum PM$: Jumlah masing masing skor DNA
- $\sum PM$: Total DNA berdasarkan jenis malware
- Dilakukan pembulatan ke bawah jika < 0.5
- Dilakukan pembulatan ke atas jika ≥ 0.5

berdasarkan output yang ditampilkan, hasil klasifikasi malware LMclicker.exe membuktikan bahwa dengan menggunakan class.sh, LMclicker dapat terdeteksi sebagai trojan dengan skor kecocokan 17 dengan *pattern assembly* yang sudah dikategorikan sebagai trojan.

Kemudian modul plotly pada python akan memvisualisasikan hasil klasifikasi berdasarkan output yang didapat. Berikut ini adalah visualisasi yang ditampilkan :



Gambar 4. Bar chart hasil proses klasifikasi



Gambar 5. Radar chart hasil proses klasifikasi

Hasil visualisasi yang ditampilkan pada gambar 4 dan gambar 5 menunjukkan bahwa LMclicker lebih berpotensi sebagai trojan dibandingkan dengan malware jenis lainnya. Untuk hasil pengujian sample LMclicker.exe, NIMDA.exe, dan petya.bin disajikan pada tabel berikut ini:

No	Nama sample	pattern assembly	Skor	Total
LMclicker.exe	Ransomware	cryptolocker.DNA	18	15
		cryptowall.DNA	17	
		matsnu.DNA	10	
		satana.DNA	15	
		wannacry.DNA	16	
	Trojan	Artemis.DNA	18	17
		bechiro.DNA	15	
		DarkTequila.DNA	17	
		Kovter.DNA	22	
		NSIS.DNA	15	
	Virus/Worm	Hybris.DNA	19	15
		klez.DNA	15	
		slammer.DNA	10	
		stuxnet.DNA	18	
		Swen.DNA	15	
NIMDA.exe	Ransomware	cryptolocker.DNA	21	21
		cryptowall.DNA	18	
		matsnu.DNA	21	
		satana.DNA	22	
		wannacry.DNA	26	
	Trojan	Artemis.DNA	18	17
		bechiro.DNA	17	
		DarkTequila.DNA	14	
		Kovter.DNA	23	
		NSIS.DNA	14	
	Virus/Worm	Hybris.DNA	18	33
		klez.DNA	26	
		slammer.DNA	79	
		stuxnet.DNA	32	
		Swen.DNA	13	
Petya.bin	Ransomware	cryptolocker.DNA	26	23
		cryptowall.DNA	25	
		matsnu.DNA	25	
		satana.DNA	22	
		wannacry.DNA	19	
	Trojan	Artemis.DNA	24	18
		bechiro.DNA	16	
		DarkTequila.DNA	15	
		Kovter.DNA	16	
		NSIS.DNA	19	
	Virus/Worm	Hybris.DNA	20	20
		klez.DNA	19	
		slammer.DNA	23	
		stuxnet.DNA	17	
		Swen.DNA	21	

Berdasarkan informasi yang disajikan oleh TABEL IV, jika *pattern assembly* hasil ekstraksi suatu malware memiliki banyak relasi yang sama dengan *pattern assembly* malware pada database maka skor penilaian akan semakin besar.

Kemudian, berdasarkan informasi yang disajikan oleh TABEL IV berbukti bahwa berdasarkan hasil proses klasifikasi dari 3 sample, beberapa malware memiliki kemiripan antara satu dengan yang lainnya walaupun dengan skor kecil. Untuk jenis malware dari sample malware yang diujikan berdasarkan referensi menunjukkan bahwa LMclicker.exe merupakan malware jenis trojan [9], NIMDA.exe merupakan malware jenis worm [10], dan petya.bin merupakan malware jenis ransomware [11].

Dengan demikian, berdasarkan referensi malware yang ada dan hasil pengujian yang tersaji pada TABEL IV membuktikan klasifikasi malware dapat dilakukan menggunakan metode relasi *pattern assembly* dan output yang dihasilkan sesuai dengan harapan.

- Proses penilaian identik:

Pada proses ini digunakan sample malware yang sudah teridentifikasi sehingga program akan memeriksa nilai identik antara setiap *pattern assembly* hasil ekstraksi dengan *pattern assembly* malware yang sudah ada pada database. Berikut ini adalah salah satu contoh proses klasifikasi malware dengan mengambil sample stuxnet.exe, command yang dijalankan adalah:

```
bash class.sh
```

output yang ditampilkan class.sh dijalankan untuk klasifikasi malware adalah sebagai berikut :

```
masukkan nama executebale :
stuxnet.exe

masukkan nama output DNA malware :
stuxnet

memulai proses pengklasifikasian...

=============================================
          ransomware
---------------------------------------------
ransomware/cryptolocker.DNA
22
ransomware/cryptowall.DNA
17
ransomware/matsnu.DNA
16
ransomware/satana.DNA
23
ransomware/wannacry.DNA
28
---------------------------------------------
skoring berdasarkan DNA ransomware:
21
=============================================


=============================================
          trojan
---------------------------------------------
trojan/Artemis.DNA
22
trojan/bechiro.DNA
16
trojan/DarkTequila.DNA
21
trojan/Kovter.DNA
16
trojan/NSIS.DNA
17
```

```
---------------------------------------------
skoring berdasarkan DNA trojan:
18
=============================================


=============================================
          virus / worm
---------------------------------------------
virus/Hybris.DNA
18
virus/klez.DNA
21
virus/slammer.DNA
31
virus/stuxnet.DNA
174
-------------------------< terdapat blok identik
virus/Swen.DNA
17
---------------------------------------------
skoring berdasarkan DNA virus/worm:
52
=============================================

++++++++++++++++++++++++++++++++++++++++++++

skor penilaian
---------------
ransomware : 21
trojan     : 18
virus/wrom : 52

kesimpulan :
lebih dominan virus

++++++++++++++++++++++++++++++++++++++++++++
```

Berdasarkan output diatas, membuktikan bahwa jika proses klasifikasi dengan malware yang *pattern assembly* nya sudah tersimpan pada database, maka program akan menampilkan informasi "terdapat blok identik" sebagai tanda bahwa malware yang diujikan sama dengan yang sudah ada pada database.

IV. KESIMPULAN DAN SARAN

Hasil dari eksperiment pada penelitian ini digunakan untuk membuktikan hipotesis yang terdapat pada bagian justifikasi.

Hipotesis pertama yaitu : Dengan memanfaatkan *pattern assembly* malware dapat digunakan untuk klasifikasi malware. Terbukti dapat dilakukan berdasarkan hasil klasifikasi pada TABLE IV bagian eksperimen.

Hipotesis kedua yaitu: Beberapa malware pasti memiliki kemiripan antara satu dengan yang lainnya walaupun dengan skor kecil. Terbukti berdasarkan TABEL IV terdapat beberapa malware yang memiliki kemiripan dengan sample ketika dilakukan percobaan klasifikasi pada bagian eksperimen.

Dengan terbuktinya hipotesis pertama dan kedua membuktikan bahwa metode klasifikasi malware berdasarkan *pattern assembly* suatu malware dapat dilakukan dan output yang dihasilkan sesuai dengan yang diharapkan. Adapun beberapa aspek yang dapat di kembangkan untuk penelitian berikutnya adalah dengan menambah dan memperkaya *pattern assembly* suatu malware agar proses klasifikasi memiliki output yang lebih akurat.

REFERENCES

[1] Impe. Koen Van, "How to Choose the Right Malware Classification Scheme to Improve Incident Response". https://securityintelligence.com/how-to-choose-the-right-malware-classification-scheme-to-improve-incident-response. 2020.

[2] Kaspersky, "Types of Malware". https://www.kaspersky.com /resource-center/threats/malware-classifications. 2020.

[3] Frank A. Coutelieris, Kanavouras. Antonios, "Experimentation Methodology for Engineers". Springer. 2018.

[4] Namanya. Anitta Patience, et all. "TheWorld of Malware: An Overview". IEEE 6th International Conference on Future Internet of Things and Cloud. 2018.

[5] Eilam, Eldad, "Reversing: secrets of reverseengineering". John Wiley & Sons. 2005.

[6] Richard L. Halterman. "Fundamentals Python Programming". Southern Adventist University.2018.

[7] Chet Ramey, Brian Fox, "Reference Documentation for Bash Edition 5.0, for Bash Version 5.0.", Free Software Foundation, Inc, 2019.

[8] Nykamp DQ, "Relation definition." From *Math Insight*. http://mathinsight.org/definition/relation. Diakses pada tahun 2020.

[9] Doctor web, "Laporan malware Agustus 2013". https://free.drweb-av.it/show/?i=3885&lng=it. Diakses pada tahun 2020.

[10] The virus Encyclopedia, "Worm". http://virus.wikidot.com/worm. Diakses pada tahun 2020.

[11] Wikipedia, "Petya (malware)". https://en.wikipedia.org/wiki/Petya_(malware). Diakses pada tahun 2020.