ISSN : 1411-1624
e-ISSN: 2621-8089

KLASIFIKASI MALWARE DENGAN MENGGUNAKAN *RECURRENT NEURAL NETWORK*

Desi Efriyani [1], Febriyanti Panjaitan[2]
Universitas Bina Darma
Jalan Jenderal Ahmad Yani No.3 Palembang
Sur-el: desiefriyani7@gmail.com [1], febriyanti_panjaitan@binadarma.ac.id[2]

Abstract : *The development of technology today has undergone many changes that are quite fast and rapid. Along with the development of technology, the internet is used as a tool that can help in solving complex problems to be practical, and easy. The increase in internet users makes crimes that use technology also increase due to rampant cybercrime activities. One of the cybercrimes used by attackers is malicious software or what is often called malware. Malware is a malicious program created to damage or break into a software or operating system, wiretapping, gaining computer access rights without the knowledge and permission of the owner, manipulating bank transactions for profit, theft of personal data, financial loss and damage to the reputation of the organization. The Recurrent Neural Network (RNN) method is a network that has a fedback link, so that the resulting output network can be used as additional input for further resistance. Recurrent Neural Netwok (RNN) is a method that can recognize data patterns well and provide accurate predictions for classifying malware and non-malware (normal files) types using the N-grams feature. This study resulted in an accuracy of 86% and an F1 Score of 85% from the total data of 215 malware and non-malware data.*

Keywords: *Classification, Malware, Deep learning, Recurrent Neural Network* (RNN).

Abstrak : *Perkembangan teknologi saat ini telah banyak mengalami perubahan yang cukup cepat dan pesat. seiring dengan perkembangan teknologi, internet dijadikan sebagai alat yang dapat membantu dalam menyelesaikan masalah yang rumit menjadi praktis, dan mudah. Meningkatnya pengguna internet membuat kejahatan yang memanfaatkan teknologi juga semakin meningkat karena maraknya kegiatan cybercerime. Cybercrime yang digunakan oleh penyerang salah satunya malicious software atau yang sering disebut malware. Malware merupakan suatu program jahat yang diciptakan untuk merusak atau membobol suatu software atau sistem operasi, penyadapan, mendapatkan hak akses komputer tanpa sepengetahuan dan izin pemiliknya, memanipulasi transaksi bank untuk mendapatkan keuntungan, pencurian data pribadi, merugikan finansial dan merusak reputasi organisasi. Metode Recurrent Neural Netwok (RNN) adalah jaringan yang memiliki umpan balik (fedback link), sehingga jaringan output yang dihasilkan dapat dijadikan input tambahan untuk tahan selanjutnya. Recurrent Neural Netwok (RNN) adalah metode yang dapat mengenali pola data dengan baik dan memberikan prediksi yang akurat untuk mengklasifikasi jenis malware dan non-malware (normal file) dengan menggunakan fitur N-grams. Penelitian ini menghasilkan akurasi 86% dan F1 Score 85% dari jumlah data sebanyak 215 data malware dan non-malware.*

Kata kunci*: Classification, Malware, Deep learning, Recurrent Neural Network* (RNN).

1. PENDAHULUAN

Perkembangan teknologi yang cukup cepat salah satunya pada perkembangan internet yang dijadikan sebagai alat membantu dalam menyelesaikan masalah yang rumit menjadi praktis dan murah[1], sehingga pengguna internet terkadang membuat kejahatan seperti pencurian data dan penyadapan transmisi pada jaringan internet[2]. Kejahatan ini disebut *cybercrime*. *Cybercrime* adalah suatu bentuk kejahatan virtual dengan memanfaatkan media

ISSN : 1411-1624
e-ISSN: 2621-8089

komputer yang terhubung ke internet, dan mengeksploitasi komputer lain yang juga terhubung ke jaringan internet [3]. *Cybercrime* yang digunakan pengguna semakin beragam, serangan tersebut salah satunya *malicious software* atau yang lebih dikenal dengan *malware*. *Malware* adalah perangkat lunak secara eksplisit didesain untuk melakukan bentuk aktivitas serangan yang berbahaya. Terdapat enam *malware* berbahaya seperti *virus, trojan, worm, exploit, bakcoor* dan *w32*[4]. *Malware* memiliki banyak karakteristik, seperti (1) Dapat menciptakan dan memodifikasi file, (2) menggunakan pustaka yang dibangun, (3) terhubung ke internet, (4) mengubah kunci registri dan sebagainya[5].

Pada umumnya pengguna internet dengan media komputer tidak sadar bahwa komputer yang digunakan telah terdapat *malware*. Komputer yang terserang *malware* akan melambat, tidak merespon dengan cepat ketika digunakan, bahkan komputer tidak bekerja dengan benar, *folder* dan *file* atau *icon* hilang dari desktop, termasuk aplikasi yang telah ter*install*. Setelah menyadari bahwa komputer yang digunakan telah terserang *malware* kebanyakan pengguna mencari *software* anti-virus yang dapat menghapus dan menghentikan serangan tersebut.

Permasalahan dari *malware* lebih berbahaya bagi instansi pemerintahan dan organisasi dibandingkan untuk pengguna pribadi. Serangan *malware* sering kali masuk melalui *email*, hasil *download* (.exe), program-program yang sudah terinfeksi dengan *malware.* Beragam tujuan yang dilakukan oleh pelaku untuk melakukan aktivitas berbahaya yang dapat merugikan orang lain, seperti penyadapan, mendapatkan hak akses tanpa sepengetahuan dan izin dari pemiliknya, memanipulasi transaksi bank untuk mendapatkan keuntungan, pencurian data pribadi, dan kerugian finansial[6].

Berdasarkan permasalahan tersebut perlu dilakukan klasifikasi *malware* yang datanya diambil dari dataset *malware* agar memudahkan dalam mempelajari dan membedakan jenis *malware*. *Reccurent Neural Network* (RNN) atau jaringan saraf berulang mampu menyimpan memori dan *feedback loop* yang memungkinkan untuk mengenali pola data dengan baik, kemudian akan melakukan prediksi yang akurat[7]. Metode RNN digunakan dalam melakukan klasifikasi *sample* data kedalam dua kelas yaitu kategori sebagai *malware* diidentifikasi dengan angka 1 (satu) dan *non-malware* diidentifikasi dengan angka 0 (nol) yang berdasarkan ciri-ciri persamaannya.

RNN adalah salah satu model yang mampu mengakomodasi *output* jaringan menjadi *input* jaringan syaraf, dan salah satu jaringan syaraf yang memiliki *feedback link* (umpat balik), sehingga jaringan *output* yang dihasilkan dapat dijadikan *input* tambahan untuk tahapan selanjutnya. RNN kemudian akan memproses data sekuensial sepanjang pola input yang henda dikenali[8].

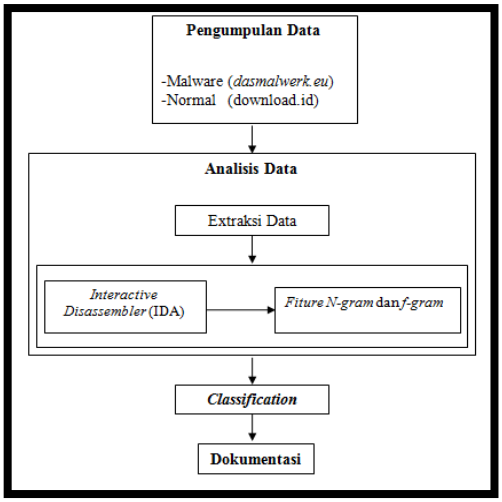
2. METODOLOGI PENELITIAN

Tahapan-tahapan dalam penelitian ini yang berfungsi untuk memecahkan suatu permasalahan agar pelaksaan penelitian sesuai

ISSN : 1411-1624
e-ISSN: 2621-8089

dengan tujuan. Metode yang digunakan dalam penelitian ini adalah menggunakan metode deskriptif yaitu metode yang dilakukan untuk menggambarkan secara sistematik dan akurat fakta dan karakteristik mengenai populasi atau bidang tertentu[9].

Berdasarkan metode tersebut maka penulis melakukan membuat alur penelitian yang diantaranya (1) pengumpulan data, (2) analisis data (3) Klasifikasi, (4) Dokumentasi.



Gambar 1. Alur Penelitian

2.1 Pengumpulan Data

DasmalWerk merupakan kumpulan *sampel malware* terbaru yang dapat di unduh melalui DasmalWerk dibuat oleh rober@artandhack.se. DasmlWerk adalah situs agar pada peneliti dapat mengambil sampel *malware* secara aman dan beresonansi. *File* yang terdapat pada situs ini berbahaya maka untuk para peneliti agar berhati-hati.

Download.id adalah website yang menyediakan *software* yang *freeware* (Perangkat lunak komputer berhak cipta yang gratis digunakan tanpa batasan waktu) dan *sofware* yang ada di *website* dapat di *download* secara gratis tanpa harus *popup* atau *spyware.*

Pada tabel 1. Terdapat Data *malware* yang telah didownload.

Tabel 1. Dataset

No.	*Nama Data*	*Jumlah data*
1.	*Backdoor*	*21*
2.	*Exploit*	*20*
3.	*Worm*	*29*
4.	*Trojan*	*20*
5.	*Virus*	*27*
6.	*W32*	*28*
7.	*Normal*	*145*
	Total	*290*

2.2 Analisis Data

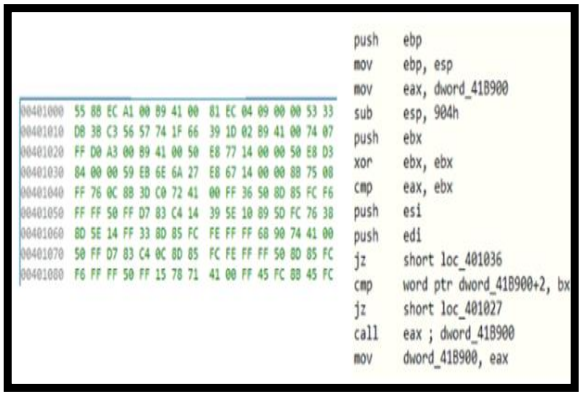
Tahapan ini dilakukan ekstraksi data, data *malware* yang telah di *download.* Ekstraksi data ini adalah tahapan proses untuk melakukan seleksi atau proses pengambilan data yang tidak diperlukan, berikut tahapan kegiatan tersebut :

1. *Interactive Diasassembler (IDA) pro*

Hasil dari training IDA Pro menghasilkan *file* normal dan *malware*.

a. Normal *Set Representation*

Dengan kode operasional *assembly* yang berbeda dengan *file malware.*



Gambar 2 . Tampilan *Hexadecimal* dan kode operasional *file* normal

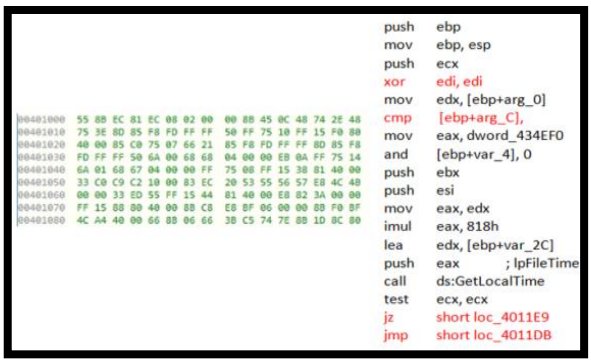
ISSN : 1411-1624
e-ISSN: 2621-8089

Saat melakukan proses IDA pro ada *file* yang tidak bisa dibaca atau dijalankan pada *software* tersebut. Setelah kode operasional bahasa *assembly* didapatkan kemudian disimpan untuk semua kode operasionalnya menjadi per*folder*. 290 data awal coba dijalankan pada IDA pro dan hanya mendapatkan 215 data setelah selesai melakukan pembongkaran pada IDA pro

b. *Malware Representation*

Pada gambar 3 terdapat bagaimana perbedaan konten dalam *malware* dan *file normal* yang hanya dijumpai dalam *file malware* saja, berikut penjelasan gambar 3 :

1) Pada gambar 3 kode xor yang digunakan adalah (xor edi, edi) yang merupakan bagian dari proses *decrypt* virus *malware* yang sudah di deklarasi pada bagian awal *script.*
2) Kode operasional pada *malware* (cmp [ebp+arg_C],) kode operasional membandingkan kode "ebp" dengan "edi" dan akan memverifikasi kode manakah yang akan diteruskan untuk ke sesi berikutnya pada proses *malware* yang dijalankan.
3) Pada kode operasional *malware* letak kode "jnz" dan "jz" sebelum dan sesudah kode operasional test ecx, ecx jz short loc_4011E9 jmp short loc_4011DB . Kode ini dapat diartikan bahwa jika lokasi tujuan tidak sesuai makan kode berikutnya akan pindah ke lokasi yang akan di set sebagai *breakpoint* dan akan diteruskan ke kode selanjutnya.



Gambar 2 . *Hexadecimal* dan Kode Operasional *Malware*

2. *Feature* N-gram

N-gram merupakan proses yang dilakukan dengan mengambil rangkai *substring* sejumlah (rangkaian token sepanjang n), N-gram sering digunakan dalam teknik analisis statik dan juga bahasa *assembly*. N-gram digunakan untuk pengolahan kode operasional bahasa *assembly* yang sudah kita dapatkan dari hasil melakukan pembongkaran data dengan menggunakan IDA pro sebelumnya.

Hasil *assembly* atau *string* yang kemudian data *string* akan diseleksi menjadi satu set tumpang tindih dengan N-grams. Manfaat menggunakan N-gram dapat menghitung data berapa frekuensi kata *string* dari data N-gram. Banyak penelitian yang menggunakan fitur N-gram yang menyarankan 4-gram untuk menjadi terbaik. Dalam penelitian ini digunakan fitur N-gram dengan range n=1, n=2, n=3. Dari hasil fitur N-gram ini dapat dilihat pada tabel 2.

ISSN : 1411-1624
e-ISSN: 2621-8089

Tabel 2. Hasil N-gram

No	*Freq*	*N-grams = 1*	*No*	*Freq*	*N-grams = 2*	*No*	*Freq*	*N-grams 3*
1	35	Mov	1	6	push esi	1	3	mov ebp esp
2	23	Push	2	3	mov ebp	2	2	mov ecx esi
3	16	Call	3	3	mov eax	3	2	xor ebx edx
4	12	Test	4	2	dec edi	4	2	xor ebx ecx
5	4	Cmp	5	2	imul edx	5	2	mov ecx edi
6	4	Imul	6	2	cmp esi	6	2	test ecx ecx
7	2	Dec	7	2	test ecx	7	2	xor edi edi
8	2	Jnb	8	1	movzx eax	8	2	mov edx ecx
9	2	Lea	9	1	lea edx	9	2	mov esi ecx
10	1	Sub	10	1	pop esi	10	1	mov ecx hwnd
11	1	Jbe	11	1	push edx	11	1	mov edx esi
12	1	Movzx	12	1	push eax	12	1	test eax eax

Setelah *file* berhasil dibongkar dan menghasilkan kode operasional, N-grams digunakan untuk mengurutkan hasil ekstrasksi kode operasional yang muncul dalam *file malware* dan *file normal* dengan mengabaikan lokasi, *memory* dan register. Contoh mov dword_434F60[eax] akan dinormalisasikan dengan mengurutkan kode operasional "mov" berdasarkan frekuensi banyaknya "mov" muncul di *file malware* dan *file normal* dengan mengabaikan lokasi, *memory* dan *register* yang hanya berlaku jika N-grams menggunakan *range* n=1.

Setelah melakukan *prepocessing* data yang awalnya 290 dan data tersebut telah dibongkar kemudian dilakukan seleksi data string (kode operasional) untuk mendapatkan f-grams menjadi 215 data.

3. Klasifikasi Data

Dalam penelitian ini, klasifikasi data menggunakan *software pyhton* dan untuk menjalankan *script/code* algoritma RNN dalam melakukan proses learning menggunakan *pycharm*. Selama *fase learning,* algoritma mempelajari pengetahuan tentang kelas yang dikenali yaitu kode operasional yang didapatkan dari hasil *assembly*.

Dari awal data 290 setelah dilakukan ekstraksi data menjadi 215 kemudian data tersebut dibagi menjadi data *testing* dan data *training*. Selama melakukan proses untuk data *learning,* proses klasifikasi mempelajari dataset *assembly* yang sudah ada data sebagai *malware* dan non-*malware (normal file)*. Kelas yang telah dibagi menjadi 2 (dua) kelas yaitu *malware* dengan kode 1 (satu) dan non-*malware (normal file)* dengan kode 0 (nol). Dari dari data (*backdoor, exploit, normal, virus, w32 dan worm*) yang didapat dari hasil *assembly* dapat dilihat bahwa hasil yang signifikan dari jumlah kode operasional yang berada pada data *malware* dan data non-*malware*. "Mov", "Push", "Cmp" merupakan kode operasional yang paling sering muncul di antara kedua file, tetapi yang membedakan adalah pada kode operasional

ISSN : 1411-1624
e-ISSN: 2621-8089

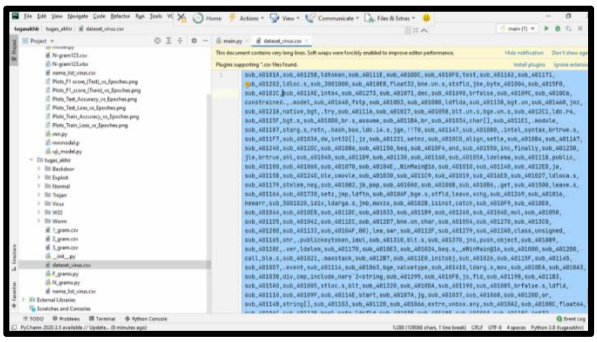
"test" dan "lea" yang memiliki ciri-ciri persamaan yang ada di file *malware* tapi tidak terdapat pada non-*malware*.

4. Dokumentasi

Dokumentasi menyimpan hasil keluaran data dari pengolahan tiap harapan proses dari *sample malware* dan non-*malware*.

3. HASIL DAN PEMBAHASAN

Pada gambar 3 terlihat bahwa data yang sudah dilakukan pembongkaran file di *software* IDA Pro. Ada 215 data yang tersimpan dalam folder dataset_virus.csv.



Gambar 3. Hasil Ektraksi Data Pada IDA Pro

Data *training* atau data latih adalah salah satu bagian penting pada proses klasifikasi terutama jika data tersebut untuk sistem pendekteksi *malware*, sedangkan, data *testing* merupakan data setelah proses training dilakukan pada mesin learning. Tahap selanjutnya untuk menentukan perfoma algoritama *recurrent neural network* yang akan diuji. 215 data dibagi 10% berarti 21% untuk testing 79% untuk *training*. Pembagian data training dan testing dapat dilihat pada gambar 4 .

Gambar 4. Hasil *Training* dan *Testing*

Error training adalah kesalahan pelatihan atau kesalahan prediksi klasifikasi model pada data yang sama dengan model yang dilatih. Sedangkan, *error testing* adalah kesalahan pengujian data dengan menggunakan dua kumpulan data yang benar-benar terpisah satu untuk melatih model dan yang lainnya untuk menghitung kesalahan klasifikasi. Dataset pertama disebut data latih dan yang kedua, data uji. Namun yang terpenting adalah mengetahui seberapa performa recurrent neural network dengan mengukur nilai akurasi. Mengukur keakuratan model dapat melihat performa terbaik sehingga RNN yang digunakan untuk melakukan pengklasifikasian lebih akurat. Dalam analisis statistik klasifikasi biner, F-score atau F-measure adalah ukuran akurasi suatu tes.

Dapat dilihat gambar 4 dapat dilihat hasil *loss/error training* dengan 1000 pengulangan preksi hasil untuk *loss/error training* berada 0.322, akurasi *training* 0.839 dan F1 *score training* 0.835 bahwa hasil dari data *training* sudah baik. Sedangkan untuk hasil akhir *error testing* 2.740, akurasi *testing* 0.619 dan F1 *score testing* 0.681 bahwa hasil akhir dari data *testing* sudah baik

ISSN : 1411-1624
e-ISSN: 2621-8089

DAFTAR PUSTAKA

[1] N. Zalavadiya and P. Sharma, "A Methodology of Malware Analysis, Tools and Technique for windows platform–RAT Analysis," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 5, no. 3, pp. 5042–5054, 2017.

[2] F. Panjaitan and R. Syafari, "Pemanfaatan Notifikasi Telegram Untuk Monitoring Jaringan," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 10, no. 2, pp. 725–732, 2019.

[3] T. H. A. Gregory, "Ketenaran Cybercrime di Indonesia," *Makal. STIMIK Perbanas*, 2005.

[4] T. A. Cahyanto, V. Wahanggara, and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *Justindo, J. Sist. Teknol. Inf. Indones.*, vol. 2, no. 1, pp. 19–30, 2017.

[5] F. Ferdiansyah, "Analisis aktivitas dan pola jaringan terhadap eternal blue dan wannacry ransomware," *JUSIFO (Jurnal Sist. Informasi)*, vol. 2, no. 1, pp. 44–59, 2018.

[6] A. P. Aldya, N. Widiyasono, and T. P. Setia, "Reverse Engineering untuk Analisis Malware Remote Access Trojan," *J. Edukasi dan Penelit. Inf.*, vol. 5, no. 1, p. 40, 2019.

[7] J. W. G. Putra, "Pengenalan Konsep Pembelajaran Mesin dan Deep Learning," *Tokyo. Jepang*, 2019.

[8] L. Chen, X. Pan, Y.-H. Zhang, M. Liu, T. Huang, and Y.-D. Cai, "Classification of widely and rarely expressed genes with recurrent neural network," *Comput. Struct. Biotechnol. J.*, vol. 17, pp. 49–60, 2019.

[9] T. Soendari, "Metode Penelitian Deskriptif," *Bandung, UPI. Stuss, Magdal. Herdan, Agnieszka*, vol. 17, 2012.

.