KLASIFIKASI *MALWARE* DENGAN *RECURRENT NEURAL NETWORK*

Desi Efriyani[1], Febriyanti Panjaitan[2], Muhamad Akbar[3], Aan Restu Mukti[4]
Fakultas Ilmu Komputer, Universitas Bina Darma
Email: desiefriyani7@gmail.com

ABSTRACT

Current technological developments have undergone many changes that are quite fast and rapid. along with technological developments, the internet is used as a tool that can help in solving complex problems that are practical, easy, easy and others. The increasing number of internet users has made crimes that utilize technology also increase due to rampant cybercerime activities. One of the cybercrime 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, stealing personal data, harming finances and damaging the reputation of the organization. The Recurrent Neural Netwok (RNN) method is a network that has a feedback link (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 properly and accurately recognize data patterns to complete malware and normal file classifications. This study resulted in an accuracy of 86% and an F1 score of 85% of the total data of 215 malware data and normal files.

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

ABSTRAK

Perkembangan teknologi saat inii 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, mudah, mudah dan lain-lainnya. 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 akurat untuk menyelesaikan klasifikasi malware dan normal file. Penelitian ini menghasilkan akurasi 86% dan F1 Score 85% dari jumlah data sebanyak 215 data *malware* dan file normal.

Kata Kunci: Klasifikasi, *Malware, Deep learning, Recurrent Neural Network* (RNN)

1. PENDAHULUAN

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, mudah, murah dan lain-lan [10]. Meningkatnya pengguna internet membuat kejahatan yang memanfaatkan teknologi juga semakin meningkat, karena maraknya kegiatan *cybercrime* yang bisa mencuri data dan penyadapan transmisi pada jaringan internet [7].

Cybercrime yang digunakan oleh penyerang semakin beragam. Serangan tersebut diantaranya *malicious software* atau yang lebih dikenal *malware*. *Malware* merupakan perangkat lunak yang secara *eksplisit* didesain untuk melakukan bentuk aktivitas serangan yang berbahaya[18]. Terdapat enam *malware* berbahaya seperti *Virus, Trojan, Worm, Exploit, Backdoor, dan W32* [5].

Malware memiliki banyak karakteristik. Misalnya, (1). Dapat menciptakan dan memodifikasi file, (2). Menggunakan pustaka yang dibangun, (3). Terhubung ke internet, (4). Mengubah kunci registri dan sebagainya [4]. Pada umumnya, user tidak sadar bahwa komputernya terkena *malware*.

Komputer yang terserang *malware* akan melambat, tidak merespon dengan cepat ketika di klick seperti mengklick *icon* dan *software icon*, bahkan komputernya tidak bekerja dengan benar, folder dan file atau *icon* hilang dari desktop, termasuk nama aplikasi yang sudah di instal. Setelah pengguna sadar bahwa komputernya terserang malware kebanyakan user mencari *software* anti-virus yang dapat menghapus dan menghentikan serangan tersebut.

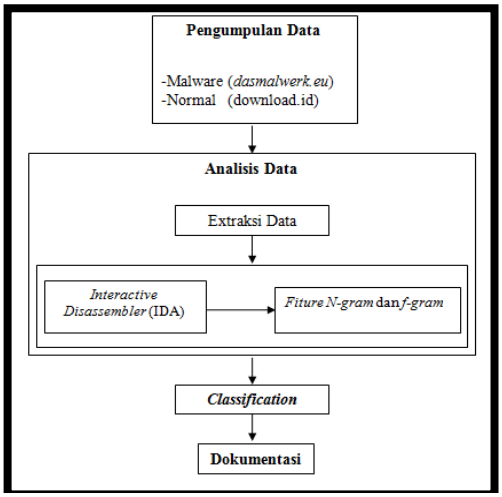
Permasalahan dari *malware* lebih berbahaya bagi intansi pemerintahan dan organisasi dibandingkan untuk pengguna pribadi. Serangan *malware* sering kali masuk melalui email, hasil download sebuah program di internet *(exe)*, program-program yang sudah terinfeksi *malware* berbahaya. Beragam tujuan yang dilakukan oleh pelaku untuk melakukan aktivitas berbahaya yang dapat merugikan orang lain seperti, penyadapan, mendapatkan hak akses komputer tanpa sepengetahuan dan izin pemiliknya, memanipulasi transaksi bank untuk mendapatkan keuntungan, pencurian data pribadi, kerugian finansial dan merusak reputasi organisasi [1].

Berdasarkan permasalahan tersebut perlu dilakukan klasifikasi malware yang datanya diambil dari dataset malware [7] agar memudahkan dalam mempelajari dan membedakan jenis malware. Reccurent Neural Network (RNN) atau jaringan bersaraf berulang [3]. RNN mampu menyimpan memori dan ingatan (feedback loop) yang memungkinkan untuk mengenali pola data dengan baik, kemudian menggunakannya untuk membuat prediksi yang akurat[8]. Metode RNN digunakan untuk membantu dalam melakukan klasifikasi sampel data kedalam dua kelas yaitu kelas kategori malware sebagai indetifikasinya dengan angka 1 dan non malware sebagai identifikasinya angka 0 yang berdasarkan ciri-ciri persamaanya.

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

2. METODOLOGI PENELITIAN

Metodologi penelitian adalah tahapan-tahapan dalam penelitian yang berfungsi untuk memecahkan suatu permasalahan agar pelaksanaan penelitian sesuai dengan tujuan. Metode yang digunakan dalam penelitian adalah metode *deskriptif*. Metode *deskriptif* adalah metode yang dilakukan untuk menggambarkan secara sistematik dan akurat fakta dan karakteristik mengenai populasi atau bidang tertentu [2]. Tahapan alur penelitian dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

2.1 Pengumpulan Data

DasmalWerk merupakan kumpulan sampel malware terbaru yang dapat diunduh. DasmalWerk dibuat oleh robert@artandhacks.se.

DasmalWerk.eu adalah situs agar para peneliti dapat peneliti sampel dengan cara yang aman dan beresonansi. File yang terdapat pada situs ini berbahaya maka untuk para peneliti harus lebih berhati-hati.

Download.id adalah website yang menyediakan *Software* yang *Freeware* (Perangkat lunak komputer berhak cipta yang gratis digunakan tanpa batasan waktu) dan *software* yang ada diwebsite dapat di download secara gratis tanpa *popup* atau *spyware* [6]. Pengumpulan data dapat dilihat pada tabel 1.

Tabel 1. Pengumpulan Data

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**

Tabel 3. Data

2.2 Analisis Data

Pada tahapan ini dilakukan ektraksi data dimana data yang sudah diambil melalui beberapa tahapan. Berikut tahapan yang akan di lalui.

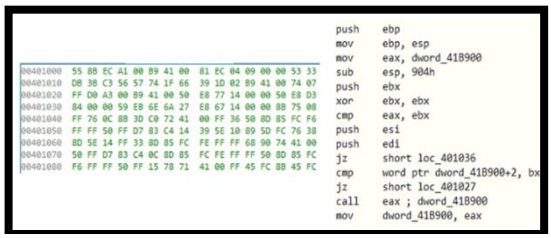
2.3 Ektraksi Data

Ektraksi data adalah tahapan proses untuk mengseleksi atau proses pengambilan data dari kata-kata yang tidak diperlukan serta kata-kata yang tidak memiliki makna.

a. Interactive Disassembler (IDA) pro

Berikut hasil dari *Diassemble file* pada dataset *malware* dan normal *file*:

1. Normal *Set Representation*



Gambar 2. Tampilan *Hexadecimal* dan Kode Operasional Normal File

2. *Malware Representation*

Gambar 3. *Hexadecimal* dan Kode Operasional *Malware*

Dilihat kedua gambar diatas dapat dijelaskan bahwa hasil traning di IDA Pro, file normal menghasilkan kode operasional *assembly* yang berbeda dengan file *malware.*

Dari gambar 3.2 diatas dapat dijelaskan bagaimana perbedaan konten dalam *malware* dan file normal yang hanya dijumpai dalam file *malware* saja, sebagai berikut.

1) Pada gambar di atas 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* digambar 5 yaitu: (cmp [ebp+arg_C],) kode operasional tersebut 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 digambar 5 letak kode jnz dan jz sebelum dan sesusah kode operasional test ecx, ecx jz short loc_4011E9 jmp short loc_4011DB kode ini dapat diartikan bahwa jika lokasi tujuan tidak sesuai, kode berikutnya akan pindah ke lokasi yang akan di *set* sebagai *breakpoint* dan akan meneruskan ke kode berikutnya.

Saat melakukan proses di IDA Pro ada file yang tidak bisa dibaca atau dijalankan pada *software* tersebut. Setelah kode operasional bahasa *assembly* didapatkan kita simpan semua kode operasionalnya perfolder. 290 data awal dicoba dijalankan pada *software* IDA pro hanya dapat 215 data setelah selesai melakukan pembongkaran di IDA Pro.

b. *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 pengelohan kode operasional bahasa *assembly* yang sudah kita dapatkan dari hasil melakukan pembongkaran data dengan menggunakan IDA Pro sebelumnya.

Hasil *assembly* yang di dapatkan oleh IDA Pro adalah berupa kumpul-kumpulan kode operional bahasa *assembly* atau *string* yang kemudian data *string* akan diseleksi menjadi satu set tumpang tindih dengan n-grams. Manfaat menggunakan n-gram hal itu dapat data menghitung berapa frekuensi kata *string* yang (kode operasional) dari data n-gram.

Banyak penelitian yang menggunakan fitur n-gram telah menyarankan 4-gram untuk menjadi yang terbaik[16]. Dalam penelitian ini digunakan fitur n-gram dengan range n = 1, n = 2, n= 3. Dan hasil dari fitur N-gram ini dapat di lihat pada tabel berikut.

Tabel 2. Hasil Fitur N-gram = 1

No	Freq	N-grams = 1
1	35	Mov
2	23	Push
3	16	Call
4	12	Test
5	4	Cmp
6	4	Imul
7	2	Dec
8	2	Jnb
9	2	Lea
10	1	Sub
11	1	Jbe
12	1	Movzx

Tabel 3. Hasil Fitur N-gram = 2

No	Freq	N-grams = 2
1	6	push esi
2	3	mov ebp
3	3	mov eax
4	2	dec edi
5	2	imul edx
6	2	cmp esi
7	2	test ecx
8	1	movzx eax
9	1	lea edx

10	1	pop esi
11	1	push edx
12	1	push eax

Tabel 4. Hasil Fitur N-gram = 3

No	Freq	N-grams = 3
1	3	mov ebp esp
2	2	mov ecx esi
3	2	xor ebx edx
4	2	xor ebx ecx
5	2	mov ecx edi
6	2	test ecx ecx
7	2	xor edi edi
8	2	mov edx ecx
9	2	mov esi ecx
10	1	mov ecx hwnd
11	1	mov edx esi
12	1	test eax eax

Setelah file berhasil dibongkar dan menghasilkan kode operasional (*opcodes*), N-grams digunakan untuk mengurutkan hasil ekstraksi 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 akan berdasarkan frekuensi banyaknya mov muncul di file *malware* dan file normal dengan mengabaian lokasi, memory dan register hanya berlaku jika N-gram menggunakan *range* n=1

Setelah melakukan Preprocessing yang data awalnya 290 dan data tersebut di bongkar dan melakukan seleksi data *string* (kode operasional) untuk mendapatkan nilai f-gram menjadi 215 data.

2.4 *Classification*

Dalam penelitian ini, klasifikasi menggunakan *python* dan untuk menjalankan *script/code* algoritma rnn dalam melakukan proses learning menggunakan *pycharm*. Selama *fase learning*, algoritma mempelajari pengetahuan tentang kelas yang dikenali adalah kode operasional yang di dapatkan dari hasil *assembly*.

Dari dawal 290 setelah dilakukan bongkar dan ektraksi data sehingga yang didapat 215 data akhir. Dari data 290 dibagi untuk data *testing* dan *training* adalah 10 yang berarti untuk data testingnya 21% dan untuk data trainingnya 79%. Selama melakukan proses untuk data learning klasifikasi mempelajari dataset *assembly* yang sudah ada data *malware* dan normal untuk kelas menentukan yang tersedia. Kelas yang tersedia ada dua kelas, *malware* dilambangkan dengan 1 dan normal dilambangkan dengan 0. Dari dari data (*backdoor, exploit, normal, virus, w32 dan worm*) yang didapat dari hasil *assembly* dapat dilihat signifikan dari jumlah kode operasional yang berada pada data *malware* dan data normal. Mov, Push, Cmp merupakan kode operasional yang paling sering muncul di antara kedua file, tetapi yang membedakan adalah pada kode operasional

test dan lea yang memiliki ciri-ciri persamaan yang ada di file *malware* tapi tidak terdapat pada file normal menurut jurnal [9].

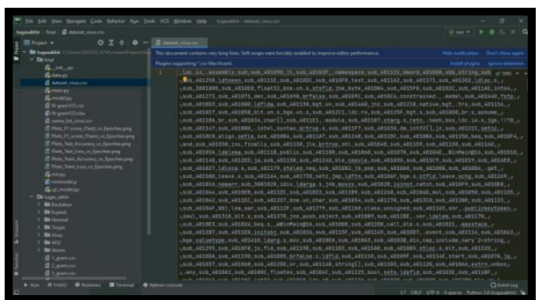
2.5 Dokumentasi

Dokumentasi menyimpan hasil keluaran data dari pengolahan tiap harapan proses dari *sample malware* dan non *malware* untuk kemudian diterapkan pada laporan skripsi.

3. HASIL DAN PEMBAHASAN

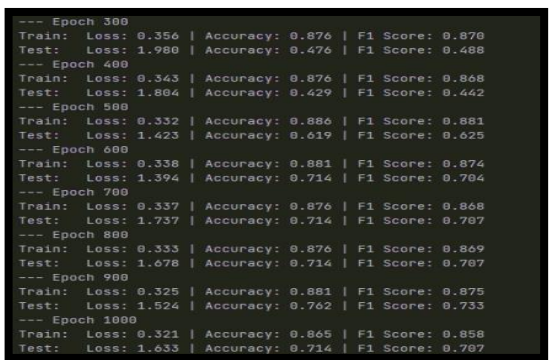
3.1 Sampel Data

Dari gambar 4 dapat dilihat data yang sudah dilakukan pembongkaran file di *software* IDA Pro. Ada 215 data yang tersipam dalam folder dataset_virus.csv.



Gambar 4. 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*. Data latih digunakan dalam teknik pembelajaran, seperti untuk melakukan proses bisnis. 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 di bawah ini.



Gambar 5. 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 5 dapat dilihat hasil *loss/error training* dengan 1000 pengulangan preksi hasil untuk *loss/error training* berada 0.325, akurasi *training* 0.865 dan F1 *score training* 0.858 bahwa hasil dari data *training* sudah baik. Sedangkan untuk hasil akhir *error testing* 1.633, akurasi *testing* 0.714 dan F1 *score testing* 0.707 bahwa hasil akhir dari data *testing* sudah baik.

4. KESIMPULAN

Adapun kesimpulan yang didapatkan dari penelitian ini adalah sebagai berikut:

1) Klasifikasi *malware* menggunakan *Recurrent Neural Network* dengan bahasa *Python* berhasil berdasarkan nilai akurasi 86% dan nilai f1 *Score* 85%.
2) Dari hasil pengklasifikasi dapat disimpulkan *malware* dilambangkan 1 dan normal file dilambangkan 0.

DAFTAR PUSTAKA

[1] Aldya, A. P., Widiyasono, N., & Setia, T. P. (2019). Reverse Engineering untuk Analisis Malware Remote Access Trojan. *JEPIN (Jurnal Edukasi dan Penelitian Informatika) Volume 5 Nomor 1, April 2019* .

[2] Azwar, S. (2014). *Metode Penelitian.* Yogyakarta: April 2014.

[3] Cho, K., & Bahdanau, D. (2014). Learning Phrse Representations using RNN Ecoder-Decoddder for Statistical Machine Translation. *Conference on Empirical Methods in Natural Language Processing (EMNLP)* .

[4] Ferdiansyah. (2018). Analisis Aktivitas dan Pola Jaringan Terhdap Eternal Blue & Wannacary Ransomware. *JURNAL SISTEM INFORMASI Volume 4, Nomor 1, Juni 2018* .

[5] https://dasmalwerk.eu/

[6] *https://download.id/*

[7] Panjaitan, F., & Syafari, R. (2019). PEMANFAATAN NOTIFIKASI TELEGRAM UNTUK MONITORING JARINGAN. *Jurnal SIMETRIS, Vol. 10 No. 2 P-ISSN: 2252-4983, E-ISSN: 2549-3108* , 726.

[8] Putra, J. W. (2019). *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning Edisi 1.3.* Tokyo, Jepang: Tokyo, Jepang.

[9] S, E. L. (2016). Klasifikasi Malware Trojan Ransomware Dengan Algoritma Support Vector Machine (SVM). *http/ars.ilkom.unsri.ac.id ISBN: 979-587-626-0* , 123.

[10] Zalavadiya, N., & Sharma, P. (2017). A Methodology of Malware Analysis, Tools and Technique for windows platform – RAT Analysis. *International Journal of Innovative Research in Computer and Communication Engineering* .