BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil penelitian mengenai sistem pendeteksian *malware*, dapat disimpulkan bahwa:

1. Pada tahap analisis dapat disimpulkan bahwa teknik analisis dinamis bisa digunakan untuk mendeteksi aktifitas *malware* yang sedang berjalan. Hal ini dapat dilihat di bagian 3.3.

2. Perekaman aktifitas dari proses yang berjalan dapat dicapai dengan cara meng-*hook* proses yang bersangkutan. Dengan perekaman ini aktifitas dari *malware* pun dapat dideteksi dan dapat diproses lebih lanjut. Hal ini dapat dilihat di bagian 3.2.

3. Pendeteksian *malware* yang dilakukan dengan menggunakan program yang dibangun dianggap cukup baik karena dari 7 percobaan fungsional memiliki tingkat deteksi 100%, dan dari 5 pengujian eksperimental memiliki tingkat deteksi 80%, dikarenakan adanya satu *malware* yang tidak terdeteksi. Hal ini dapat dilihat di bagian 5.3.

4. Pengiriman data ke server tentang aktifitas *malware* yang berjalan di sebuah sistem sudah berjalan dengan baik. Hal ini dapat dilihat di bagian **??**.

5. Program pendeteksi aktifitas tersembunyi yang dibuat sudah memenuhi tujuan dari penelitian yang dilakukan yaitu membuat aplikasi yang dapat mendeteksi aktifitas *malware* di komputer dan mengirimkan data tentang aktifitas yang terjadi di komputer.

6.2 Saran

Berdasarkan kesimpulan yang telah dipapar idi atas, maka penulis dapat memberikan saran diantaranya:

1. Program jika dibangun di atas bahasa C++, akan berjalan lebih cepat. Pada saat ini, ketika program dijalankan, program yang dihook akan mengalami *delay* sekitar 1 sampai 2 detik. Hal ini disebabkan bahasa pemrograman C# tidak bisa langsung melihat ke memori, sedangkan bahasa pemrograman C++ bisa langsung melihat ke memori. Sehingga bahasa pemrograman C++ dianggap bisa lebih cepat dari C#.

DAFTAR REFERENSI

[1] Elisan, C. (2012) *Malware, Rootkits & Botnets A Beginner's Guide* Beginner's Guide. McGraw-Hill Education.

[2] LLC, C., Mitchell, M., Samuel, A., dan Oldham, J. (2001) *Advanced Linux Programming* Landmark. Pearson Education.

[3] Father, H. (2004) Hooking windows api - technics of hooking api functions on windows. *CodeBreakers-Journal*, **1**.

[4] R, V. dan Raii, N. (2012) Windows api based malware detection and framework analysis. *International Journal of Scientific and Engineering Research*, **3**.

[5] Aycock, J. (2006) *Computer Viruses and Malware*. Springer US.

[6] Ki, Y., Kim, E., dan Kim, H. K. (2015) A novel approach to detect malware based on api call sequence analysis. *Int. J. Distrib. Sen. Netw.*, **2015**, 4:4–4:4.

[7] Richter, J. (1999) *Programming Applications for Microsoft Windows*. Microsoft Press.

[8] Damodaran, A., Troia, F. D., Visaggio, C. A., Austin, T. H., dan Stamp, M. (2017) A comparison of static, dynamic, and hybrid analysis for malware detection. *J. Computer Virology and Hacking Techniques*, **13**, 1–12.

[9] Moser, A., Kruegel, C., dan Kirda, E. (2007) Limits of static analysis for malware detection. *In 23rd Annual Computer Security Applications Conference (ACSAC), IEEE Computer*. Society Press. USA.