BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Database signature malware yang mengimplementasi *fuzzy hash* dengan metode *signature-based detection* dapat meningkatkan jumlah deteksi *malware* dengan rata-rata peningkatan 31,84% dan rata-rata peningkatan akurasi sebesar 16,63% terhadap *hash* SHA256. Pembangunan empat buah *database signature* yang dilakukan dalam penelitian dengan program *Database Signature Builder* menunjukkan perbandingan ukuran *database* yang mengimplementasi *fuzzy hash* 7% − 12% lebih besar dari *database* yang mengimplementasi *hash* SHA256. Dalam delapan skenario uji deteksi menggunakan program *Malware Detector*, menunjukkan waktu yang dibutuhkan untuk menyelesaikan operasi *scanning* menggunakan *database signature* yang mengimplementasi *fuzzy hash* 200% − 500% lebih lambat dari *database* yang menggunakan *hash* SHA256.

Rata-rata peningkatan jumlah deteksi *database fuzzy hash* terhadap *database* SHA256 yang dicapai berdasarkan hasil uji deteksi dalam delapan skenario pengujian adalah 31,84%, dengan minimal 0% dan maksimal 97,09%. Peningkatan 97,09% dicapai pada nilai batas toleransi kemiripan minimal (*threshold*) 25%. Namun, pada *threshold* tersebut, terjadi penurunan tingkat akurasi deteksi hingga -17,73%. Rata-rata peningkatan akurasi deteksi *database fuzzy hash* terhadap *database* SHA256 yang dicapai berdasarkan hasil uji deteksi dalam delapan skenario pengujian adalah 16,63%, dengan minimal -17,73% dan maksimal 97,09%. Peningkatan nilai *threshold*, berpengaruh pada penurunan

jumlah deteksi. Dalam penelitian ini menunjukkan, peningkatan jumlah deteksi dan akurasi optimal dapat dicapai pada *threshold* 50%.

5.2 Saran

Berikut beberapa saran terkait penelitian yang bermanfaat untuk penelitian selanjutnya.

1. Diperlukan optimasi penghitungan *fuzzy hash* dan optimasi pendeteksian *signature-based* lebih lanjut untuk mengurangi waktu yang dibutuhkan dalam proses deteksi.
2. *Task* dalam aplikasi dilakukan menggunakan sebuah *BackgroundWorker* sederhana sehingga masih terdapat *lag* pada aplikasi, terutama bila dilakukan dua atau lebih proses deteksi secara bersamaan. Diperlukan optimasi *multi-threading* yang lebih baik.
3. Aplikasi *Malware Detector* dalam penelitian ini digunakan untuk pengujian deteksi *malware* tanpa *removal malware*. Untuk melengkapi fungsinya, diharapkan penelitian selanjutnya dapat mengimplementasi *removal* untuk setiap *malware* yang ada dalam *database*.
4. Aplikasi *Malware Detector* yang dibangun hanya memiliki kemampuan *on-demand scanner* dan deteksi hanya berdasarkan *signature*-nya. Diharapkan penelitian selanjutnya, dapat mengimplementasi kemampuan *real-time protection* dan metode deteksi dilengkapi dengan *behavioural-based detection*, yaitu analisis perilaku *malware* pada saat *runtime*.
5. Pengembangan aplikasi *Database Signature Builder* terintegrasi ke dalam *Malware Detector* sehingga pengguna dapat menambahkan sendiri *malware* ke dalam *database* (semacam *blacklist*) lebih mudah.

6. Sampel *malware* yang digunakan dalam penelitian ini hanya terbatas pada *malware executable* dan *scripting* pada sistem operasi Windows. Untuk pengujian implementasi *fuzzy hash* yang lebih luas, diharapkan penelitian selanjutnya, dapat menggunakan sampel *malware* yang lebih beragam untuk berbagai jenis sistem operasi.
7. Dalam pengambilan sampel, diharapkan melakukan *preprocessing* terlebih dahulu terhadap sampel *malware* terutama *malware* yang berjenis *script* (*plaintext*) untuk mengetahui potensi *false positive* dengan *file plaintext* bersih lainnya.