

PENGUNAAN METODE CYCLIC REDUNDANCY CHECK 32 (CRC32) SEBAGAI PENDETEKSIAN KERUSAKAN FILE DOKUMEN

***¹Reza Alamsyah, ²Mhd. Dicky Syahputra Lubis**

STMIK Methodist Binjai; Jl. Jend. Sudirman No. 136 Binjai – Telp: 061-88742021
Teknik Informatika

e-mail: ^{*1}reza@stmikmethodistbinjai.ac.id, ²mhddicky@stmikmethodistbinjai.ac.id

Abstrak

Dokumen-dokumen digital dihasilkan dengan menggunakan aplikasi-aplikasi pengolah kata seperti Microsoft Word atau Notepad menjadi sebuah file dengan ekstensi yang berbeda sesuai dengan aplikasi pengolah kata yang digunakan. Seperti halnya dokumen yang bersifat manual, dokumen-dokumen digital ini tidak terlepas dari bahaya kerusakan. Salah satu kerusakan yang sering terjadi pada dokumen digital adalah perubahan yang terjadi didalam dokumen tersebut akibat dari serangan virus atau pemadaman komputer dengan paksa. CRC32 (Cyclic Redundancy Check 32 Bit) merupakan suatu metode yang menggunakan fungsi hash dalam membaca sebuah struktur dalam sebuah file dalam transmisi atau penyimpanan sebuah data. CRC32 dapat digunakan untuk mendeteksi error (kerusakan) pada sebuah data yang mungkin terjadi pada saat transmisi data atau pengiriman data. Perangkat lunak yang dirancang mampu mendeteksi apakah suatu file mengalami perubahan atau kerusakan berdasarkan nilai checksum dan nilai CRC dari file tersebut. Perangkat lunak ini memiliki kelemahan dalam hal proses pendeteksian kerusakan file, dimana perangkat lunak tidak dapat mendeteksi lebih dari satu file sekaligus dalam satu proses pendeteksian.

Kata kunci: CRC32, File, Word

Abstract

Digital documents are generated using word processing applications such as Microsoft Word or Notepad into a file with different extensions according to the word processing application used. As with any document that is manual, these digital documents are inseparable from the danger of damage. One of the most frequent damages to digital documents is the changes occurring within the document resulting from virus attacks or computer outages by force. CRC32 (Cyclic Redundancy Check 32 Bit) is a method that uses a hash function in reading a structure in a file in the transmission or storage of a data. CRC32 can be used to detect errors on data that may occur during data transmission or data transmission. The designed software is able to detect whether a file has changed or damaged based on the checksum value and CRC value of the file. This software has a weakness in the case of file damage detection, where the software can not detect more than one file simultaneously in one detection process.

Kata kunci: CRC32, File, Word

1. PENDAHULUAN

Salah satu aktivitas yang sangat dipengaruhi oleh perkembangan teknologi komputer adalah pencatatan data dan informasi. Dengan menggunakan komputer, pencatatan data dan informasi menjadi bersifat elektronik dalam bentuk dokumen digital sehingga lebih gampang dalam mengakses serta cepat dan penyampaian informasi yang dibutuhkan. Dokumen-dokumen digital

ini dihasilkan dengan menggunakan aplikasi-aplikasi pengolah kata seperti *microsoft word* atau *notepad* menjadi sebuah *file* dengan ekstensi yang berbeda sesuai dengan aplikasi pengolah kata yang digunakan.

Salah satu kerusakan yang sering terjadi pada dokumen digital adalah perubahan yang terjadi didalam dokumen tersebut akibat dari serangan virus atau pemadaman komputer dengan paksa. Kerusakan yang diakibatkan oleh virus maupun pemadaman komputer secara paksa ini mengakibatkan *cluster-cluster* pada dokumen digital berubah. *Cluster* yang berubah ini mengakibatkan informasi yang dibaca pada saat dokumen digital dibuka berubah dari aslinya. *Cluster-cluster* ini merupakan informasi penting yang dibutuhkan sistem untuk dapat mengenali identitas dokumen digital serta isi dari dokumen digital tersebut.

CRC32 (*Cyclic Redundancy Check 32 Bit*) merupakan suatu metode yang menggunakan fungsi hash dalam membaca sebuah struktur dalam sebuah *file* dalam transmisi atau penyimpanan sebuah data. CRC32 dapat digunakan untuk mendeteksi *error* (kerusakan) pada sebuah data yang mungkin terjadi pada saat transmisi data atau pengiriman data.

Dengan menggunakan CRC32, dapat dirancang sebuah perangkat lunak yang dapat mendeteksi kerusakan pada sebuah dokumen digital sehingga kerusakan dapat ditangani lebih dini dan tidak menyebar ke dokumen-dokumen digital yang lain.

2. METODE PENELITIAN

Dalam melakukan pendeteksian terhadap kerusakan suatu file, sebelumnya perlu melakukan analisa terhadap metode CRC32.

2.1 Analisis Sistem

Dalam melakukan deteksi terhadap kerusakan sebuah *file*, CRC 32 menggunakan nilai *checksum* sebuah *file* untuk menentukan letak kerusakan *file* tersebut. Adapun proses kerja dari CRC 32 dalam mendeteksi kerusakan sebuah *file* adalah sebagai berikut :

1. CRC 32 membaca *file* yang akan dideteksi kerusakannya untuk mengambil informasi nilai *checksum file* tersebut.
2. Nilai *checksum* ini kemudian diolah menjadi sebuah fungsi *hash* satu arah dengan nilai 32 bit. Fungsi *hash* 32 bit inilah yang akan digunakan CRC 32 sebagai identitas *file* yang dideteksi kerusakannya.
3. CRC 32 kemudian mengakses *registry* sistem untuk mengambil informasi *file* yang tercatat dalam *registry* sistem. Hal ini dilakukan untuk membandingkan nilai *checksum* yang tercatat pada *registry* sistem dengan nilai fungsi *hash* hasil pengolahan.
4. Bila nilai *checksum* dari *registry* sistem berbeda dengan nilai fungsi *hash* hasil pengolahan, maka CRC 32 akan menyatakan bahwa *file* yang dideteksi dalam kondisi rusak, tapi jika nilai *checksum* dari *registry* sistem sama dengan nilai fungsi *hash* hasil pengolahan, maka CRC 32 menyatakan *file* yang dideteksi tidak dalam kondisi rusak.
5. Bila hasil pendeteksi menyatakan bahwa *file* dalam kondisi rusak, CRC 32 akan mengolah nilai fungsi *hash* 32 bit dengan cara membaginya dengan nilai *checksum* dari *registry* sistem. Hal ini dilakukan untuk mengetahui dimana lokasi kerusakan pada *file* tersebut.

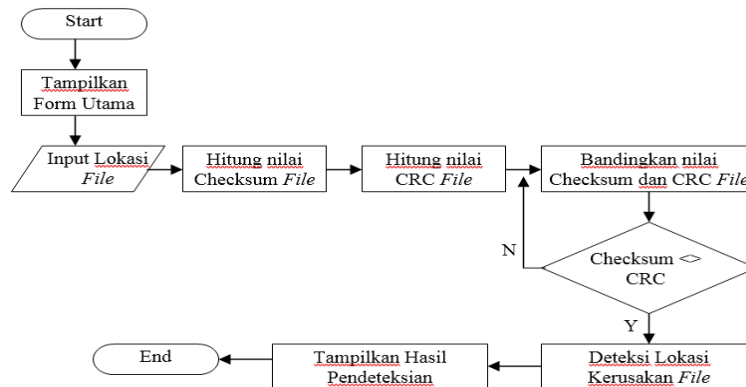
2.2. Desain

Adapun rancangan-rancangan yang penulis lakukan antara lain rancangan proses, rancangan antar muka serta rancangan struktur menu perangkat lunak.

2.2.1. Rancangan Proses

Untuk menjelaskan proses-proses yang terjadi dalam perangkat lunak pendeteksi kerusakan *file* dengan menggunakan metode CRC 32 ini, penulis menggunakan bagan alir

(flowchart). Adapun bentuk *flowchart* dari rancangan proses yang terjadi pada perangkat lunak ini seperti terlihat pada Gambar 2.1



Gambar 2.1 Rancangan Proses Flowchart

Proses kerja sistem dimulai dengan menampilkan form Utama. Sistem kemudian menunggu penginputan lokasi *file* yang akan dideteksi. Setelah lokasi *file* ditentukan, sistem kemudian melakukan penghitungan nilai *checksum* dari *file* tersebut. Dari nilai *checksum* yang diperoleh, sistem kemudian melakukan penghitungan terhadap nilai CRC *file* tersebut.

Sistem kemudian mengakses registri sistem operasi untuk mengambil nilai registri dari *file* yang dipilih *user*. Nilai registri ini kemudian dibandingkan dengan nilai CRC yang telah dihitung sebelumnya. Jika nilai *checksum* registri *file* sama dengan nilai CRC, maka sistem menyatakan bahwa *file* tersebut tidak mengalami perubahan (tidak mengalami kerusakan). Jika nilai *checksum* registri *file* tidak sama dengan nilai CRC, maka sistem menyatakan bahwa *file* tersebut telah mengalami perubahan (mengalami kerusakan).

Jika *file* mengalami perubahan, sistem kemudian melacak lokasi dimana perubahan terjadi dan menampilkan hasilnya kepada *user*.

2.3. Implementasi

Adapun bentuk implementasi yang dilakukan dalam perancangan perangkat lunak pendeteksian kerusakan *file* menggunakan CRC 32 ini berupa fasilitas yang disediakan pada perangkat lunak, kebutuhan sistem, tampilan perangkat lunak serta pengujian sistem.

2.3.1. Fasilitas Perangkat Lunak

Adapun fasilitas-fasilitas yang disediakan oleh perangkat lunak yang penulis rancang ini adalah sebagai berikut :

1. Fasilitas untuk memilih daftar file yang akan dideteksi
Fasilitas ini merupakan fasilitas yang dapat digunakan oleh pengguna untuk memilih folder dimana file yang akan dideteksi berada. Dengan fasilitas ini, pengguna dapat melakukan pengecekan terhadap seluruh file dokumen yang ada di dalam folder yang dipilih.
2. Fasilitas untuk mendeteksi kerusakan file
Fasilitas ini merupakan fasilitas yang dapat digunakan oleh pengguna untuk melihat apakah ada file dokumen yang mengalami kerusakan berdasarkan folder yang dipilih.
3. Fasilitas untuk menyimpan nilai CRC
Fasilitas ini merupakan fasilitas yang dapat digunakan oleh pengguna untuk menyimpan hasil pendeteksian file dokumen dalam bentuk sebuah daftar yang berisi nilai CRC dari setiap file yang dideteksi. Daftar ini akan menjadi acuan dalam melakukan pendeteksian

terhadap file yang sama untuk mendeteksi apakah ada perubahan dalam file tersebut dikemudian hari.

2.3.2. Kebutuhan Perangkat Lunak

Agar perangkat lunak pendeteksi kerusakan file dokumen ini dapat berjalan dengan baik, ada beberapa spesifikasi yang harus dipenuhi, antara lain :

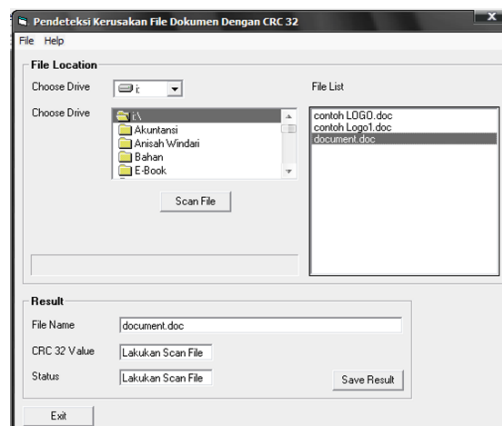
1. Spesifikasi *Hardware* (Perangkat Keras)
 - a. Micro Processor Pentium IV
 - b. Harddisk minimal 10 GB untuk tempat sistem beroperasi dan sebagai media penyimpanan data.
 - c. Memory minimal 1 GB
 - d. Monitor Super VGA
2. Spesifikasi *Software* (Perangkat Lunak)
 - a. Sistem operasi *Windows XP*
 - b. Microsoft Visual Basic 6.0
 - c. Anti Virus untuk mencegah kemungkinan terjadinya kerusakan sistem yang disebabkan oleh virus-virus yang masuk baik dari disket maupun media input lainnya

2.4. Tampilan Program

Bentuk tampilan program yang diperoleh dari hasil implementasi perancangan perangkat lunak pendeteksi kerusakan *file* menggunakan CRC 32 antara lain adalah tampilan form Utama.

1. Tampilan Form Utama

Tampilan Form Utama merupakan tampilan induk dari perangkat lunak yang dirancang. Pada form ini ditampilkan daftar *file* dokumen yang akan atau telah dideteksi kerusakannya serta menu-menu dan tombol-tombol yang dapat digunakan pengguna untuk berinteraksi dengan perangkat lunak. Adapun tampilan Form Utama seperti ditunjukkan pada Gambar 2.2



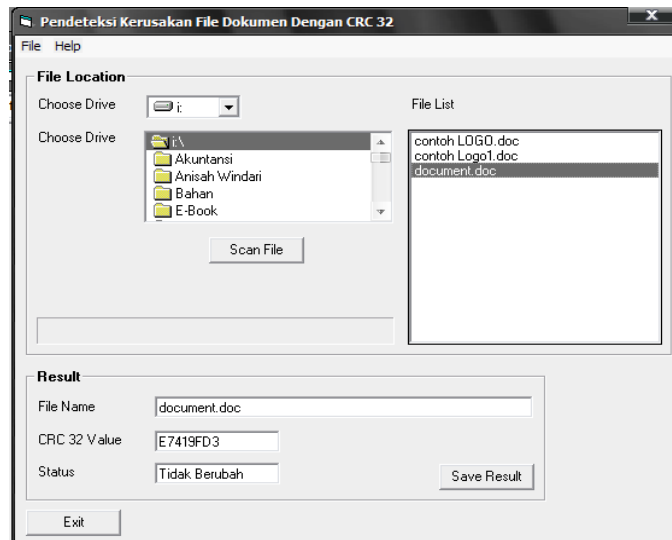
Gambar 2.2. Tampilan Form Utama

2.5. Pengujian Perangkat Lunak

Untuk memastikan bahwa perangkat lunak telah dapat berjalan dengan baik, penulis melakukan serangkaian pengujian terhadap perangkat lunak yang telah diimplementasikan. Adapun pengujian yang penulis lakukan meliputi pengujian pemilihan scan *file* serta pengujian pengecekan kerusakan.

2.5.1. Pengujian Scan File

Pengujian scan *file* penulis lakukan untuk memastikan bahwa perangkat lunak telah dapat membaca nilai CRC 32 dari *file* pada *folder* yang dipilih pengguna. Adapun hasil dari pengujian ini sebagaimana terlihat pada Gambar 2.3.



Gambar 2.3. Pengujian Scan File

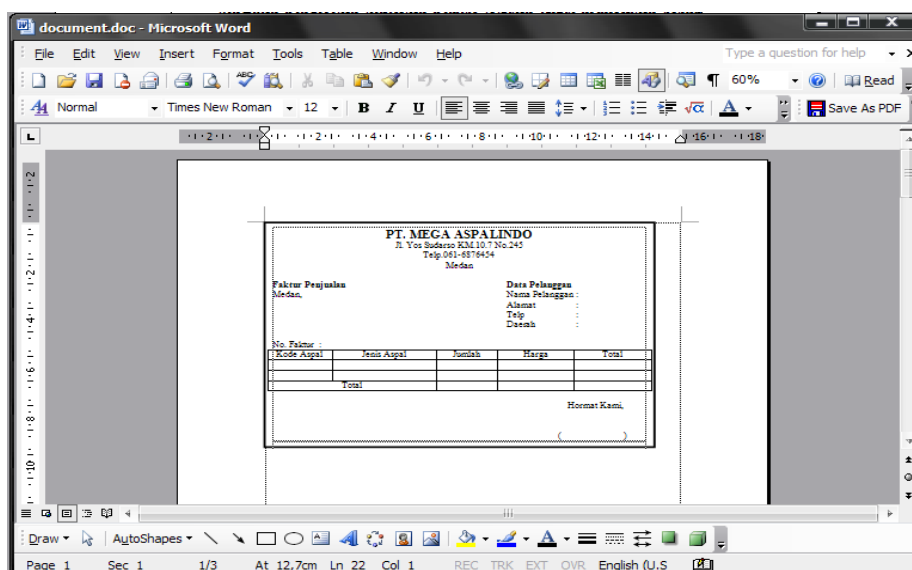
Dari hasil yang ditunjukkan pada Gambar 4.3, terlihat bahwa saat penulis memilih *file* dengan nama "document.doc", nilai yang ditampilkan pada *text box* CRC 32 Value berubah dari "Lakukan Scan File" menjadi "E7419FD3". Hasil ini menunjukkan bahwa perangkat lunak telah dapat menghitung nilai CRC 32 dari *file* "document.doc" pada saat dilakukan Scan File.

Nilai pada *text box* Status berubah dari "Lakukan Scan File" menjadi "Tidak Berubah". Nilai "Tidak Berubah" ini dihasilkan karena nilai CRC 32 dari *file* "document.doc" belum terdaftar pada *file* daftar sehingga sistem mengasumsikan bahwa nilai CRC 32 dari *file* tersebut belum mengalami perubahan.

2.5.2. Pengujian Pengecekan Kerusakan

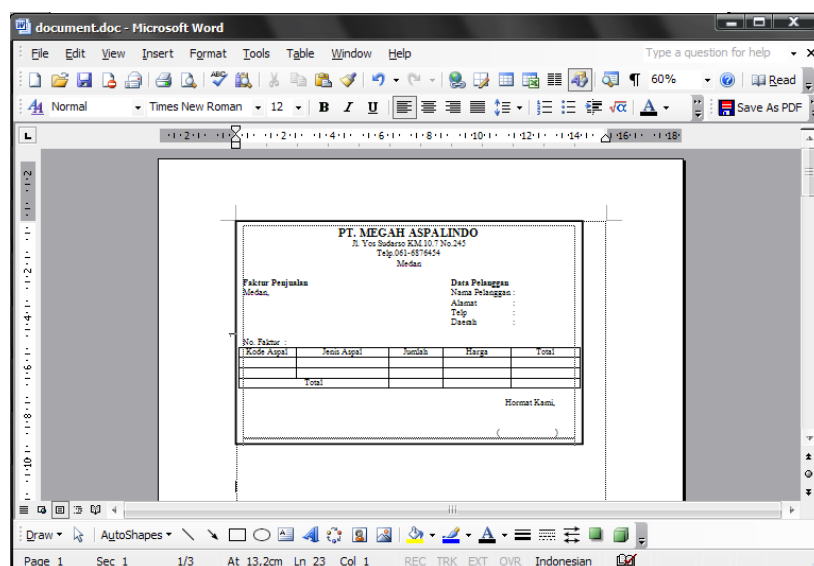
Pengujian pengecekan kerusakan penulis lakukan untuk memastikan bahwa perangkat lunak telah dapat mendeteksi setiap perubahan yang terjadi pada *file* berdasarkan hasil pendeteksian sebelumnya.

Sebelum melakukan pengujian ini, penulis melakukan pengecekan terhadap isi dari *file* "document.doc" yang hasilnya seperti terlihat pada Gambar 2.4.



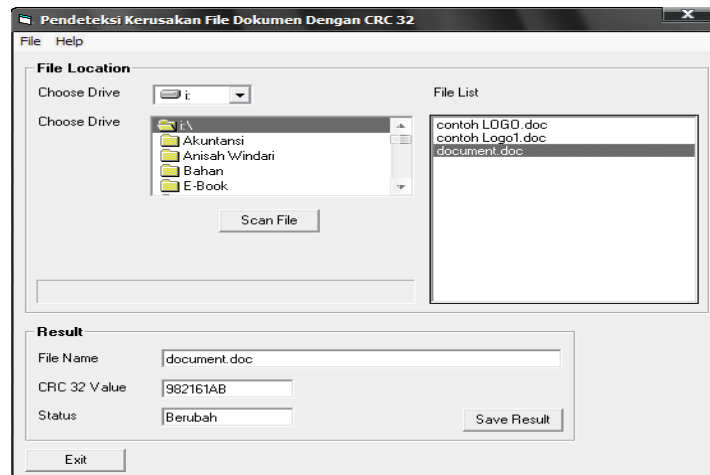
Gambar 2.4. Isi Awal Dari File Yang Diuji

Selanjutnya, penulis melakukan perubahan dari isi *file* "document.doc" tersebut dengan mengubah tulisan "PT. MEGA ASPALINDO" menjadi "PT. MEGAH ASPALINDO" seperti terlihat pada Gambar 2.5. Penulis kemudian menyimpan hasil perubahan tersebut.



Gambar 2.5. Perubahan Isi File Yang Diuji

Selanjutnya penulis melakukan Scan *File* sekali lagi untuk melihat apakah ada perbedaan antara nilai CRC 32 sebelum dan sesudah dilakukan perubahan pada *file* "document.doc". Adapun hasilnya seperti terlihat pada Gambar 2.6.



Gambar 2.6. Pengujian Pengecekan Kerusakan

Dari hasil yang ditunjukkan pada Gambar 4.7, terlihat bahwa ada perubahan pada nilai yang ditampilkan pada *textbox* CRC 32 Value, dimana nilai sebelumnya adalah “E7419FD3” berubah menjadi “982161AB”. Perubahan juga terjadi pada nilai yang ditampilkan pada *textbox* Status, dimana nilai sebelumnya adalah “Tidak Berubah”, berubah menjadi “Berubah”.

Dari hasil pengujian ini, terlihat bahwa perangkat lunak telah dapat melakukan pendeteksian kerusakan pada *file* dimana dengan melakukan sedikit saja perubahan pada *file* tersebut, sistem dapat mendeteksi adanya perubahan pada *file* berdasarkan nilai CRC 32 yang dihitung.

2.6. Kelebihan dan Kelemahan Perangkat Lunak

Berdasarkan hasil implementasi dari perancangan perangkat lunak, penulis memperoleh beberapa kelebihan dan kelemahan dari perangkat lunak ini.

Adapun kelebihan dari perangkat lunak ini adalah :

1. Perangkat lunak ini dapat mendeteksi apakah ada perubahan yang terjadi pada *file* walaupun perubahan yang terjadi sangat kecil.
2. Perangkat lunak dapat menyimpan hasil pendeteksian untuk digunakan sebagai bahan pembandingan bagi proses pendeteksian berikutnya.
3. Perangkat lunak dapat melakukan pendeteksian terhadap lebih dari satu *file* dokumen, selama *file* tersebut berada dalam satu *folder*.

Sedangkan kelemahan dari perangkat lunak ini adalah :

1. Walaupun dapat melakukan pendeteksian pada lebih dari satu *file*, perangkat lunak tidak dapat mendeteksi sekaligus lebih dari satu *file*. Hal ini akan mengakibatkan waktu pendeteksian akan lama jika dilakukan pendeteksian pada *file* dengan jumlah yang banyak.
2. Perangkat lunak membutuhkan daftar dari hasil pendeteksian sebelumnya sebelum dapat menentukan apakah terjadi perubahan pada *file* tersebut.
3. Perangkat lunak hanya dapat mendeteksi kerusakan *file* pada dokumen digital dengan format TXT dan DOC saja. Perangkat lunak belum mampu melakukan pendeteksian kerusakan pada *file* dokumen dengan form

4. KESIMPULAN

Berdasarkan hasil dari penelitian yang penulis lakukan mengenai pendeteksian kerusakan *file* dokumen dengan menggunakan metode CRC 32 yang dirancang, penulis dapat menarik kesimpulan sebagai berikut :

1. CRC 32 dapat mendeteksi perubahan pada sebuah *file*, sekecil apapun perubahan tersebut.
2. Nilai CRC 32 sebuah *file* dihasilkan dari nilai empat byte terakhir dari nilai *checksum file* tersebut dalam bentuk hexadecimal.
3. CRC 32 hanya dapat digunakan untuk menghitung nilai CRC dari sebuah *file* dengan nilai bit 32 kebawah.
4. Perangkat lunak yang dirancang dapat mendeteksi perubahan pada *file* berdasarkan daftar nilai CRC dari hasil pendeteksian sebelumnya

5. SARAN

Adapun saran yang dapat penulis berikan untuk mengembangkan perangkat yang lunak ini adalah sebagai berikut :

1. Perangkat lunak dapat dikembangkan lebih lanjut agar dapat mendeteksi kerusakan *file* pada format dokumen digital yang lain, khususnya format DOCX.
2. Perangkat lunak dapat dikembangkan lebih lanjut agar dapat melakukan pendeteksian kerusakan pada dua atau lebih *file* sekaligus sehingga dapat menghemat waktu proses pendeteksian yang dibutuhkan.
3. Untuk pengembangan lebih lanjut, perangkat lunak dapat dikembangkan agar dapat mendeteksi seluruh *file* yang terdapat dalam sebuah *drive* tidak hanya seluruh *file* yang terdapat dalam sebuah *folder* saja.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Allah SWT yang telah memberi Rezeki serta dukungan Kesehatan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Hariyanto, Bambang. 2003. *Sistem Operasi Lanjut*. Edisi Pertama. Bandung : Informatika.
- [2] Hariyanto, Bambang. 2009. *Sistem Operasi*. Bandung : Informatika.
- [3] Jain, A., Dubes, R. 1988. *Algorithms for clustering data*. New Jersey : Prentice-Hall Inc.
- [4] Jones, David T. 2003. *An Improved 64-bit Cyclic Redundancy Check for Protein Sequences*. London : University College London.
- [5] Joseph, Hammond, Jr., James E. B., Shyan-Shiang L. 1975. *Development of a Transmission Error Model and an Error Control Model*, National Technical Information Service.
- [6] Koopman, P. 2002. *32-Bit Cyclic Redundancy Codes for Internet Applications*. The International Conference on Dependable Systems and Networks. USA. Pittsburgh.

- [7] Koopman, P., Chakravarty, T. 2004, *Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks*. USA. Pittsburgh.
- [8] Kreith, Frank. 2000. *The CRC Handbook of Thermal Engineering*. CRC Press.
- [9] Narapratama, Ditto. 2006. *Perbandingan Performansi Algoritma Adler-32 dan CRC-32 pada Library Zlib*. Bandung : Institut Teknologi Bandung.
- [10] Ramabadran, T. V., Gaitonde, S. S. 1988. *A tutorial on CRC computations*.
- [11] Ross, A., and Eli B. 1996. *Tiger: A Fast New Hash Function*. Fast Software Encryption, FSE'96, LNCS 1039.
- [12] Salim, Hartojo. 1989. *Virus Komputer, teknik pembuatan & langkah-langkah penaggulangannya*. Yogyakarta : Andi OFFSET.
- [13] Setiawan, Andi. 2009. *Algoritma Adler, CRC, Fletcher dan Implementasi pada Mac*. Bandung : Institut Teknologi Bandung.
- [14] Stigge, M., Plotz, H., Muller, W., Redlich, J. 2006. *Reversing CRC – Theory and Practice*. Berlin: Humboldt University Berlin.
- [15] Tanenbaum, Andrew. 1990. *Structured Computer Organization*. Third edition. Amsterdam : Prentice-hall inc.
- [16] Wijayanto, I. S. 2006. *Penggunaan CRC32 Dalam Integritas Data*. Bandung : Institut Teknologi Bandung.
- [17] Anachriz. 1999. *CRC and how to Reverse it*. [http:// www.woodmann.com/ fravia/ crcut1.htm](http://www.woodmann.com/fravia/crcut1.htm). Diakses tanggal 2 Maret, 2010.
- [18] Anhar. 2009. *Checksum CRC32*. [http://ilmukomputer.org/wp-content/uploads/2009/06 / anharku-checksumcrc32.pdf](http://ilmukomputer.org/wp-content/uploads/2009/06/anharku-checksumcrc32.pdf). Diakses tanggal 10 Februari, 2010.
- [19] Greg, Cook. 2009. *Catalogue of parameterised CRC algorithms*. <http://regrex.bbcmicro.net/crc-catalogue.htm>. Diakses tanggal 20 Maret, 2010.
- [20] Thaler, Pat. 2003. *16-bit CRC polynomial selection*. INCITS T10. <http://www.t10.org/ftp/t10/document.03/03-290r0.pdf>. Diakses tanggal 20 Maret, 2010.