# KAJIAN CRC32 UNTUK MENDETEKSI PERUBAHAN ISI FILE DOCUMENT

Indra M. Sarkis, S
Fakultas Ilmu Komputer Universitas Methodist Indonesia
Jl. Hang Tuah no 8 Medan
poetramora@gmail.com

**Abstract**

Cyclic redundancy check 32 Bit or known by CRC32 is a technique used in data communication to check the damage information sent or received between the sender and receiver are working at the data link layer on layer of the Open Systems Interconnection (OSI) by comparing the checksum value frames are sent to the frame received during transmission takes place. CRC32 techniques applied attempted to detect changes in a file that may occur due to human vandal through virus attack. In this study CRC 32 is applied with the same technique, namely by comparing the checksum of a file obtained from the registry system with the hash function of the value of the checksum file to be compared. From the comparison of the checksums will be known whether a file is changed or not, by trying to apply it to prove results on microsoft office document file and pdf.

*Key Word :CRC32, detect , File, Document*

## I. PENDAHULUAN

Umumnya kerusakan pada file terjadi karena adanya perubahan isi dari file, yang dapat mengakibatkan file tersebut tidak dapat dibuka sama sekali bahkan file tersebut rusak.

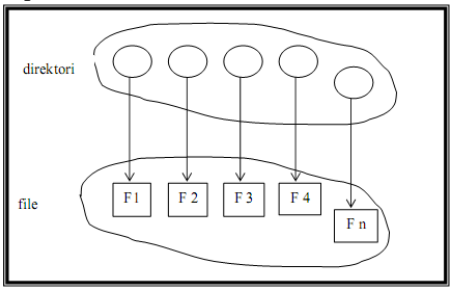
Kerusakan ini umumnya diakibatkan oleh virus maupun penyebab matinya komputer secara *upnormal* dapat mengakibatkan *cluster-cluster* pada file khususnya file dokumen berubah atau tidak sesuai dengan struktur awalnya. *Cluster* yang berubah ini mengakibatkan informasi yang dibaca pada saat file dokumen dibuka berubah dari aslinya. *Cluster-cluster* ini merupakan informasi penting yang dibutuhkan sistem untuk dapat mengenali identitas file dokumen serta isi dari file dokumen tersebut. Perubahan *cluster* ini bisa merusak isi dokumen sehingga merugikan pengguna terutama jika file dokumen tersebut merupakan dokumen penting baginya.

CRC32 (*Cyclic Redundancy Check* 32 *Bit*) merupakan suatu teknik 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. Teknik ini menghitung nilai *checksum* dari panjang bit sebuah data yang kemudian membandingkannya dengan aturan CRC dengan menggunakan kunci 32 bit untuk mendeteksi apakah data tersebut mengalami perubahan atau tidak.

## II. TINJAUAN PUSTAKA

### Organisasi Sistem File (Jogiyanto,2000)

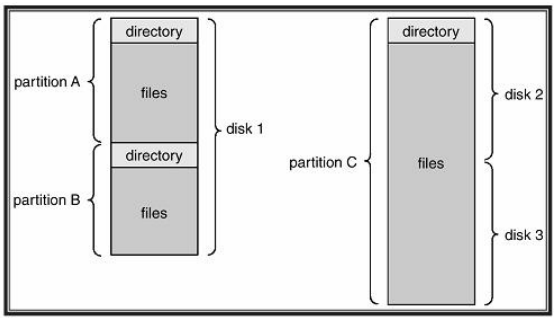
Setiap file dalam komputer tersimpan di dalam direktori. Direktori adalah kumpulan titik yang berisi informasi tentang semua file, seperti terlihat pada Gambar 1




Gambar 1 Struktur Direktori (Wijayanto,2006)

Untuk mengatur semua data dalam direktori, sistem file menggunakan organisasi yg dilakukan dalam dua bagian. Pertama, sistem file dipecah ke dalam partisi, yang disebut juga *minidisk*(IBM) atau *volume* (PC dan *Macintosh*). Setiap disk pada sistem berisi sedikitnya satu partisi, merupakan struktur *low-level* dimana file dan direktori berada. Terkadang, partisi digunakan untuk menentukan beberapa daerah terpisah dalam satu disk, yang diperlakukan sebagai perangkat penyimpan yang terpisah. Sistem lain menggunakan partisi yang lebih besar dari sebuah disk untuk mengelompokkan disk ke dalam satu struktur logika.

Kedua, setiap partisi berisi informasi mengenai file di dalamnya. Informasi ini disimpan pada entry dalam *device directory* atau *volume table of contents*. Direktori menyimpan informasi seperi nama, lokasi, ukuran dan tipe untuks semua file dari partisi tersebut. Secara umum, organisasi sistem file seperti ditunjukka pada Gambar 2.2




Gambar 2 Organisasi Sistem File
(Wijatanto,2006)

**Fungsi Hash**

*Hash function* atau fungsi hash adalah suatu cara menciptakan

*fingerprint* dari berbagai data masukan. *Hash function* akan mengganti atau mentranspose-kan data tersebut untuk menciptakan fingerprint, yang biasa disebut *hash value*. *Hash value* biasanya digambarkan sebagai suatu string pendek yang terdiri atas huruf dan angka yang terlihat *random* (data biner yang ditulis dalam notasi heksadesimal).

Suatu *hash function* adalah sebuah fungsi matematika, yang mengambil sebuah panjang variabel string input, yang disebut pre-image dan mengkonversikannya ke sebuah string output dengan panjang yang tetap dan biasanya lebih kecil, yang disebut message digest5. *Hash function* digunakan untuk melakukan *fingerprint* pada *pre-image*, yaitu menghasilkan sebuah nilai yang dapat menandai (mewakili) *pre-image* sesungguhnya.

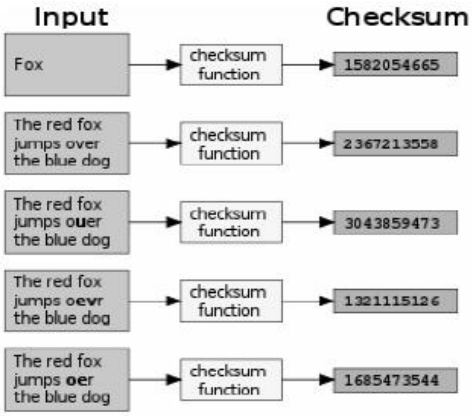
**Cheksum(Narapratama,2006)**

*Checksum* adalah teknologi untuk menandai sebuah file, dimana setiap file yang sama harus memiliki *checksum* yang sama, dan bila nilai checksumnya meskipun berbeda satu bit saja, maka file tersebut merupakan file yang berbeda walaupun memiliki nama file yang sama.

*Checksum* digunakan untuk verifikasi suatu data yang disimpan atau yang dikirim dan diterima. Setiap kali terjadi proses pengiriman data, *checksum* akan mengenali file tersebut untuk melihat apakah data yang diterima sudah sesuai dengan data yang dikirimkan. Fungsi inilah yang menjadikan *checksum* sangat efektif untuk melakukan pengecekan terhadap proses transfer suatu data.

*Checksum* akan membaca ulang, menghitung dan membandingkan file yang diterima dengan file yang ditransfer. Bila ada perbedaan nilai, maka *checksum* akan menganggap bahwa telah terjadi kesalahan, distorsi atau korupsi selama penyimpanan atau pengiriman.

Fungsi *checksum* akan selalu menghasilkan *checksum* dengan panjang yang tetap dan cukup identik satu sama lain. Dengan kata lain, bila pesan yang dimasukan berbeda, maka *checksum*-nya juga akan berbeda. Adapun bentuk dari mekanisme *checksum* seperti terlihat pada Gambar 2.3




Gambar 2.3 .Mekanisme *Checksum*
(Wijayanto,2006)

**CRC32**

CRC (*Cyclic Redundancy Check*) adalah algoritma untuk memastikan integritas data dan mengecek kesalahan pada suatu data yang akan ditransmisikan atau disimpan. Data yang hendak ditransmisikan atau disimpan ke sebuah media penyimpanan rentan sekali mengalami kesalahan, seperti halnya *noise* yang terjadi selama proses transmisi atau memang ada kerusakan perangkat keras.

CRC dapat digunakan untuk memastikan integritas data yang hendak ditransmisikan atau disimpan. CRC bekerja secara sederhana, yakni dengan menggunakan perhitungan matematika terhadap sebuah bilangan yang disebut sebagai *Checksum*, yang dibuat berdasarkan total bit yang hendak ditransmisikan atau yang hendak di simpan.

Dalam transmisi jaringan, khususnya dalam jaringan berbasis teknologi *Ethernet*, *checksum* akan dihitung terhadap setiap *frame* yang hendak ditransmisikan dan ditambahkan ke dalam *frame* tersebut sebagai informasi dalam *header* atau *trailer*. Penerima frame tersebut akan menghitung kembali apakah *frame* yang ia terima benar-benar tanpa kerusakan, dengan membandingkan nilai *frame* yang dihitung dengan nilai *frame* yang terdapat dalam *header frame*. Jika dua nilai tersebut berbeda, maka *frame* tersebut telah berubah dan harus dikirimkan ulang.

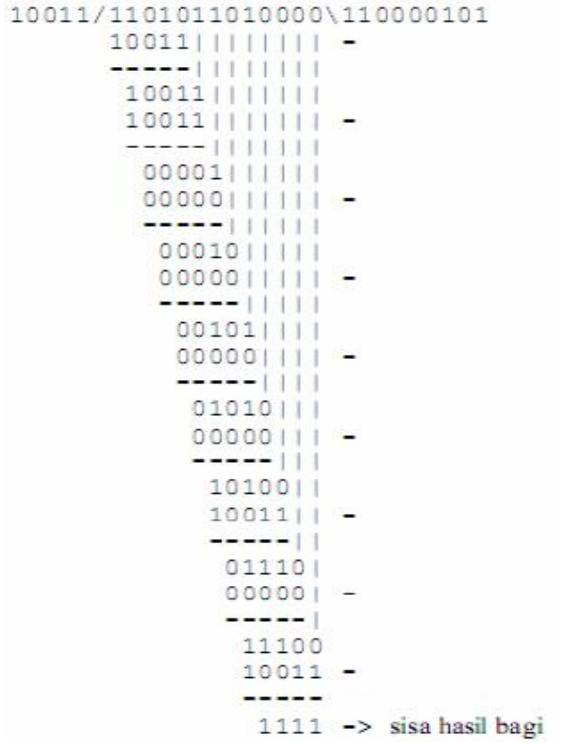
CRC didesain sedemikian rupa untuk memastikan integritas data terhadap degradasi yang bersifat acak dikarenakan *noise* atau sumber lainnya (kerusakan media dan lain-lain). CRC tidak menjamin integritas data dari ancaman modifikasi terhadap perlakukan yang mencurigakan oleh para *hacker*, karena memang para penyerang dapat menghitung ulang *checksum* dan mengganti nilai *checksum* yang lama dengan yang baru untuk membodohi penerima.

CRC32 merupakan salah satu algoritma *Cyclic Redundancy Check* yang menghasilkan

*checksum* sebesar 32 bit. Prinsip utama yang digunakan CRC32 adalah dengan melakukan pembagian polinomial dengan mengabaikan bit-bit *carry*.

CRC dihasilkan dengan membagi bilangan polinomial tersebut dengan sebuah divisor/ pembagi. Setiap operasi pembagian pasti akan menghasilkan suatu sisa hasil bagi (meskipun ada kemungkinan bernilai 0), tetapi ada perbedaan dalam melakukan pembagian pada penghitungan CRC ini. Dari nilai hasil bagi, sisa hasil bagi, dan bilangan pembagi kita bisa mendapat bilangan yang dibagi dengan mengalikan bilangan pembagi dengan hasil bagi dan menambah dengan sisa hasil bagi.

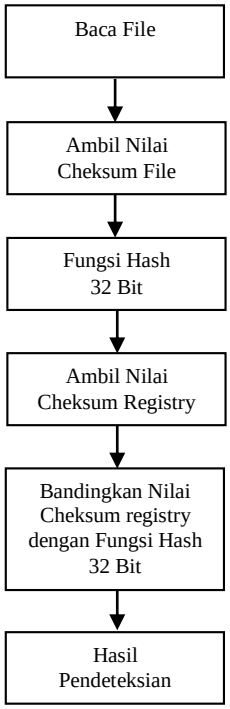
Dalam penghitungan CRC, operasi pengurangan dan penjumlahan dilakukan dengan melakukan operasi XOR pada bit-bit, jika operasi tersebut ekivalen dengan operasi pengurangan pada aljabar biasa. Perhitungan CRC juga mengabaikan bit carry setelah bit tersebut melewati suatu operasi. Adapun proses penghitungan *checksum* pada CRC32 seperti telihat pada Gambar 4 berikut.(Anhar, 2009)




Gambar 4 Proses Perhitungan Cheksum CRC 32 (Wijayanto,2006)

## III. METODE PENELITIAN

Flow proses pendeteksian perubahan file dengan CRC32, ditunjukkan pada gambar 5



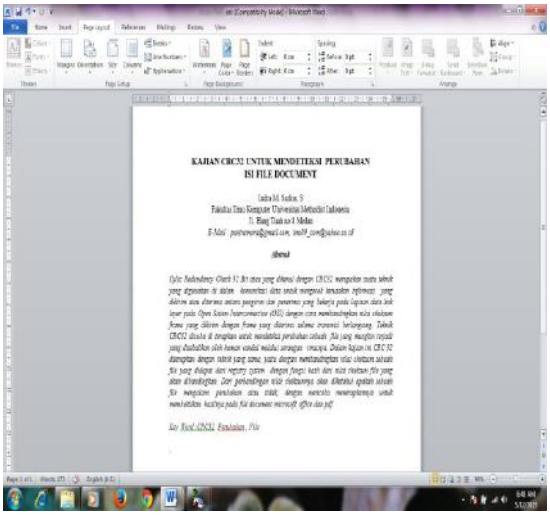

Gambar 5 Flow Proses Pendeteksian File dengan CRC32

1. CRC membaca file yang akan dideteksi perubahannya untuk mengambil informasi nilai *checksum* file tersebut.
2. Nilai *checksum* kemudian diubah menjadi fungsi *hash* satu arah dengan nilai 32 bit. Fungsi *hash* 32 bit inilah yang akan digunakan CRC32 sebagai identitas file yang dideteksi perubahannya.
3. CRC32 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 CRC32 akan menyatakan bahwa file yang di deteksi dalam kondisi berubah isinya, tapi jika nilai *checksum* dari *registry* sistem sama dengan nilai fungsi *hash* hasil pengolahan, maka CRC32 menyatakan file yang di deteksi tidak dalam kondisi berubah isinya.
5. Bila hasil pendeteksi menyatakan bahwa isi file dalam kondisi berubah, CRC32 akan menetapkan nilai fungsi *hash* 32 bit dengan cara membaginya dengan nilai *checksum* dari *registry* sistem. Hal ini dilakukan untuk mengetahui dimana lokasi perubahan isi pada file tersebut.

## IV. HASIL DAN PEMBAHASAN

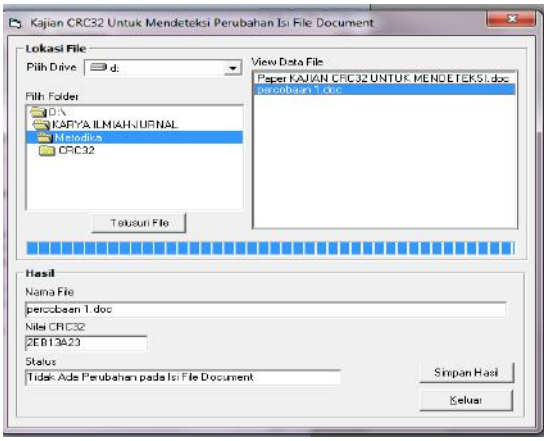
Adapun pengujian yang lakukan untuk mendeteksi perubahan isi file document ditunjukkan pada Langkah-Langkah Berikut:

1. Membuat file document microsoft word sebagai percobaan dengan nama file percobaan1.doc, adapun isi file document percobaan1.doc ditunjukkan pada gambar 6




Gambar 6 Isi File Document percobaan1.doc Sebelum diubah

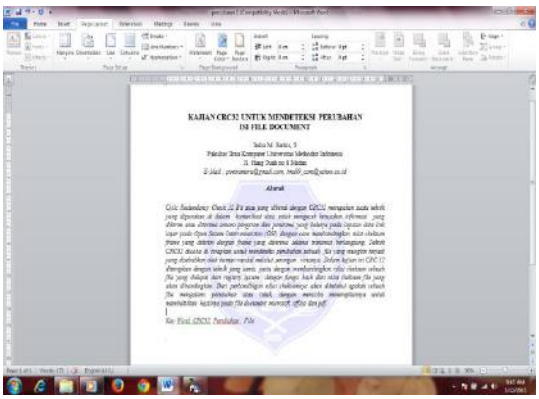
2. Melakukan proses pembacaan nilai CRC 32 dari *file* pada *folder* yang dipilih. Hasil dari pengujian untuk membaca nilai CRC32 ditunjukkan pada Gambar 7




Gambar 7 Tampilan Form Membaca Nilai CRC32

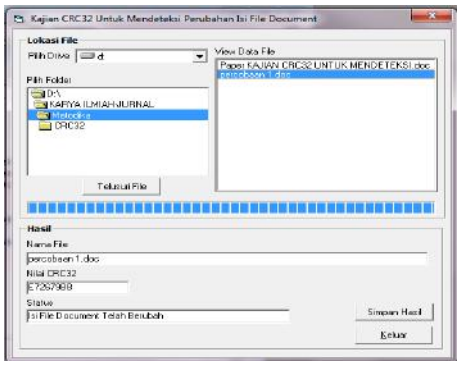
Hasil yang ditunjukkan pada Gambar 4.2, terlihat bahwa nilai CRC32 dari file percobaan1.doc setelah diclick telusuri file adalah **"2EB13A23"** dengan status "Tidak Ada Perubahan pada Isi File Document"

3. Melakukan perubahan isi file percobaan1.doc dengan menambahkan watermark pada document, sehingga tampak seperti pada gambar




Gambar 8 Isi File Document percobaan1.doc Setelah diubah

4. Melakukan proses pendeteksian perubahan isi file document, proses pendeteksian ditunjukkan pada gambar 9




Gambar 9 Tampilan Form Pendeteksian

hasil yang ditunjukkan pada Gambar 4.4, telah terdeteksi bahwa ada perubahan pada nilai CRC32, di mana nilai CRC32 sebelumnya adalah "**2EB13A23**" berubah menjadi "**E72679BB**". Dengan demikian perubahan isi file document tersebut dapat terdeteksi dengan baik. Untuk mempertegas hasil pengujian pendeteksian perubahan isi file document, penulis menguji dengan beberapa teknik yang ditunjukkan pada table berikut

Tabel 1 Hasil Pengujian Berdasarkan Perubahan Isi File Document

| Nama File Document | Nilai CRC32 Sebelum Isi Document Diubah | Nilai CRC32 Setelah Isi Document Diubah | Ket |
|---|---|---|---|
| Percobaan1.doc | **2EB13A23** | **E72679BB** | ✓ |
| Percobaan2.docx | **8BF4A54A** | **B836DA32** | ✓ |
| Percobaan3.pdf | **A342dBAF** | **D32A578C** | ✓ |
| Percobaan4.xls | **BAD8462F** | **2F3862AC** | ✓ |
| Percobaan5.ppt | **D345ACDF** | **BFA244DD** | ✓ |

Tabel 2 Hasil Pengujian Berdasarkan Perubahan Nama File Document

| Nama File Document Sebelum Diubah | | Nama File Document Setelah Diubah | |
|---|---|---|---|
| Nama File | Nilai CRC32 | Nama File | Nilai CRC32 |
| Percobaan1.doc | **2EB13A23** | Coba1.doc | **2EB13A23** |
| Percobaan2.docx | **8BF4A54A** | Coba2.docx | **8BF4A54A** |
| Percobaan3.pdf | **A342dBAF** | Coba3.pdf | **A342dBAF** |
| Percobaan4.xls | **BAD8462F** | Coba4.xls | **BAD8462F** |
| Percobaan5.ppt | **D345ACDF** | Coba5.ppt | **D345ACDF** |

## V.KESIMPULAN

Dari hasil pengujian yang dilakukan dengan teknik CRC32 terhadap beberapa file asli yang dilakukan perubahan nama file atau edit isi terhadap isi file tersebut, ditarik kesimpulan sebagai berikut:

1. Sekecil apapun perubahan yang dilakukan terhadap isi file document yang diuji, CRC32 mampu mendeteksi perubahan yang terjadi pada file tersebut dengan membandingkan nilai CRC dari hasil pendeteksian dengan nilai CRC Sebelumnya
2. Dapat disimpulkan bahwa CRC32 dapat juga digunakan untuk mendeteksi isi file yang sama walaupun nama filenya berbeda atau file duplikat

## VI. DAFTAR PUSTAKA

Anhar. (2009). Checksum CRC32. http://ilmukomputer.org/wp–content/ uploads/2009/ 06/anharku-checksumcrc32.pdf/. Diakses tanggal 09 Mei, 2015

Jogiyanto, 2000, Pengenalan Komputer, Penerbit Andi Offset, Yogyakarta.

Narapratama, Ditto. (2006). “Perbandingan Performansi Algoritma Adler-32 dan CRC-32 pada Library Zlib Bandung: Institut Teknologi Bandung

Wijayanto, 2006,”Penggunaan CRC32 Dalam Integritas Data”, Bandung: Institut Teknologi Bandung

Virus Tutorial. 2006. Mengenali Virus Lewat Checksum Error dengan metode CRC32. http://virologi.info/virologist/modules/news/article.php?storyid=6 , diakses tanggal 10 Mei 2015