PENCARIAN DAN PENGHAPUSAN FILE DUPLIKAT PADA MEDIA PENYIMPANAN KOMPUTER DENGAN ALGORITMA CRC32

Indra M. Sarkis, S

Program Studi Sistem Informasi,
Fakultas Ilmu Komputer Universitas Methodist Indonesia
Jl. Hang Tuah no 8 Medan

poetramora@gmail.com

ABSTRAK

The use of computers in general, are sometimes unaware of file storage, file format with the same type and with a different file name, the computer storage media over and over again. This often occurs when downloading files from the Internet or copy files from other storage media whose file names are different but the content of the same file. In this study CRC 32 algorithm is used as a tool or a subset to search for duplicate files by comparing files with each file to another by reading the checksum value of a file and then compare it with another file checksum value. If the checksum value of some of those files are the same, then the files are declared equal or duplicated even if the file name is different. Outcomes of this research is a software to find and remove duplicate files on computer storage.

***Key word** : Search file, delete file, duplicate files, CRC 32, Algorithm.*

1. Pendahuluan

Banyak pengguna komputer menyimpan dokumen yang sama dengan nama file yang berbeda, hal ini sering terjadi karena kurangnya perhatian para pengguna komputer terhadap pengelolaan berkas berkas atau *file-file* yang disimpan di dalam media penyimpanan. Hal ini sering terjadi dan tidak disadari oleh pengguna saat *download* file dari internet atau mencopy file dari media penyimpanan yang lain yang mengakibatkan pemborosan *space* media penyimpanan karena menyimpan dokumen atau file yang duplikat yang tidak berarti.

CRC-32 merupakan suatu teknik pengecekan(*checksum*) yang umumya digunakan di dalam Lapisan *Datalink* OSI (*Open Sistem Interconection*) untuk mendeteksi *frame-frame* yang dikirimkan atau yang diterima saat berlangsungnya komunikasi dengan cara membandingkan nilai *cheksum frame* yang dikirim dengan nilai *cheksum frame* yang diterima selama transmisi berlangsung.[5]

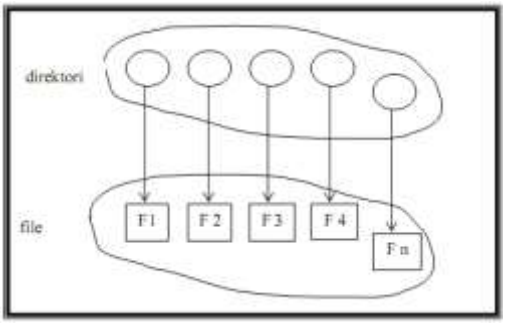
Penelitian sebelumnya penulis sudah pernah mengkaji algoritma CRC 32 dalam mendeteksi perubahan *content* pada suatu file, di mana dalam penelitian ini, CRC 32 diterapkan dengan membandingkan nilai *cheksum* sebuah file yang didapat dari *registry* sistem dengan fungsi *hash* dari nilai *cheksum* file yang akan dibandingkan. Dari perbandingan nilai *cheksum*nya akan diketahui apakah sebuah file mengalami perubahan atau tidak, dengan mencoba menerapkannya untuk membuktikan hasilnya pada file *document microsoft office* dan pdf.[2]

Pada Penelitian ini CRC32 dikaji dan diterapkan penggunaannya untuk melakukan pencarian file duplikat dengan melakukan *checksum* dari setiap file yang satu dengan file yang lainnya dan kemudian membandingkan nilai CRC-nya. Jika nilai CRC 32 sama maka dinyatakan file tersebut duplikat

2. Tinjauan Pustaka

2.1 Organisasi Sistem File [3]

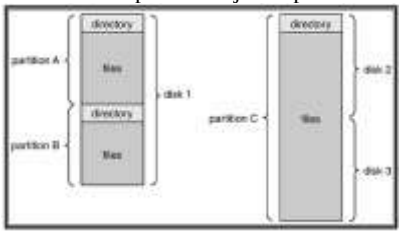
Setiap file dalam komputer tersimpan di dalam direktori. Direktori adalah kumpulan titik yang berisi informasi tentang semua file, seperti terlihat pada Gambar 2.1



Gambar 2.1 Struktur Direktori [4]

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 2.2 Organisasi Sistem File [4]

2.2 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).

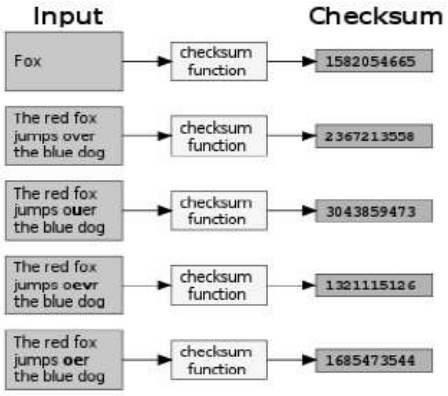
Suatu *hash function* adalah sebuah fungsi matematika, yang mengambil sebuah panjang variabel string input, yang disebut pre-image dan mengkonversikannya ke sebuah string output dengan panjang yang tetap dan biasanya lebih kecil, yang disebut message digest5. *Hash function* digunakan untuk melakukan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

2.3 Cheksum [3]

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* [4]

2. 4 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.

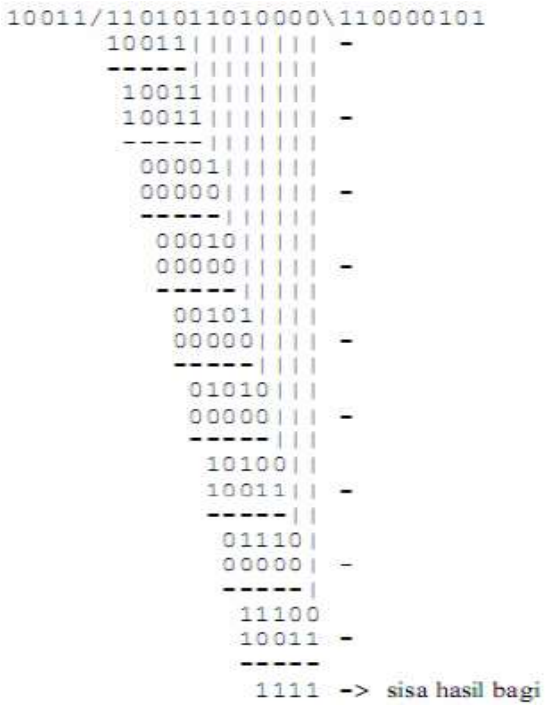
Dalam transmisi jaringan, khususnya dalam jaringan berbasis teknologi *Ethernet*, *checksum* akan dihitung terhadap setiap *frame* yang hendak ditransmisikan dan ditambahkan ke dalam *frame* tersebut sebagai informasi dalam *header* atau *trailer*. Penerima frame tersebut akan menghitung kembali apakah *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.

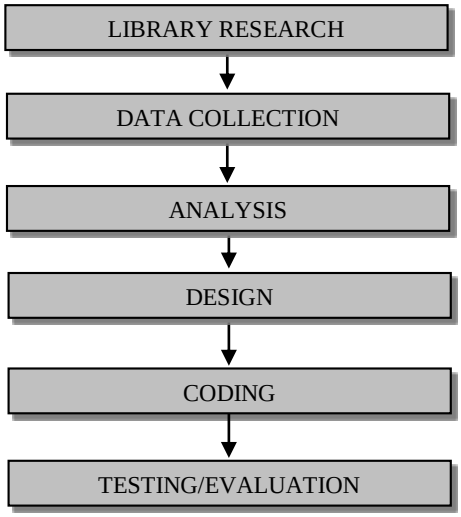
Dalam penghitungan CRC, operasi pengurangan dan penjumlahan dilakukan dengan melakukan operasi XOR pada bit-bit, jika operasi tersebut ekivalen dengan operasi pengurangan pada aljabar biasa. Perhitungan CRC juga mengabaikan bit carry setelah bit tersebut melewati suatu operasi. Adapun proses penghitungan *checksum* pada CRC32 seperti telihat pada Gambar 2.4 berikut. [1]



Gambar 2. 4 Proses Perhitungan Cheksum CRC 32 [4]

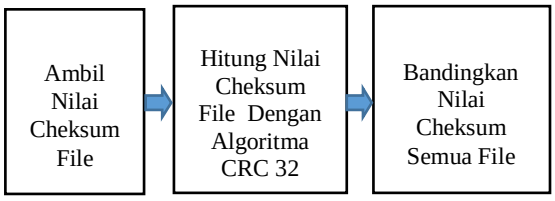
3. Metode Penelitian

pencarian dan penghapusan file duplikat pada media penyimpanan komputer dengan algoritma CRC 32 yang dikaji terhadap domain masalah yang dikaji dimulai dari *library research*, *data collection*, *analysis*, *design*, *coding* serta *testing*/*evaluation* . Untuk mencapai *goal* dari topik yang diteliti dapat dilihat *frame work* penelitian yang ditunjukkan pada gambar 3.1



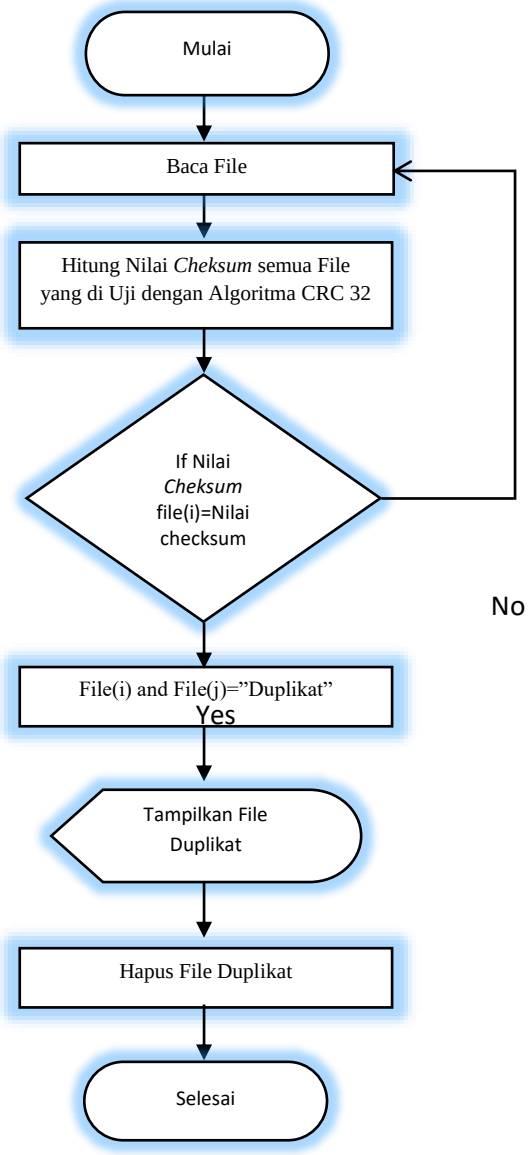
Gambar 3.1 Frame Work Penelitian

1. *Library Research*
   Mempelajari berbagai sumber pustaka seperti *journal*, *text book*, *article*, karya ilmiah dan berbagai bahan lainya dari *intenet* yang berkaitan dengan penggunaan algoritma CRC 32
2. *Data Collection*
   Mengumpulkan berbagai jenis format file yang akan digunakan sebagai sumber data pada penelitian untuk dilakukan proses pencarian dan penghapusan file duplikat dengan algoritma CRC 32.
3. Analysis
   Analisis yang dilakukan dalam pencarian dan penghapusan file duplikat dengan algoritma CRC32 sebagai berikut :



Gambar 3.2 Flow Proses Pencarian file duplikat dengan Algoritma CRC32

   a. Ambil *Cheksum* file
      Merupakan tahapan untuk mengambil nilai *cheksum* file, tahapan ini merupakan tahapan yang pertama sekali dilakukan sebelum mendeteksi file tersebut apakah ada yang duplikat dengan file lainnya. Pada penelitian ini untuk mengambil nilai *checksum* dari sebuah file digunakan fungsi kernel 32 dan win 32 pada bahasa pemrograman
   b. Hitung Nilai *Cheksum* File dengan Algoitma CRC 32
      Perhitungan nilai *checksum* sebuah file dengan algoritma CRC 32 sebagai berikut :
      - Asumsikan Nilai *Cheksum* dari sebuah File adalah 16854735
      - Konversi nilai *checksum* file ke biner
      - Kemudian dihitung dengan Algoritma CRC 32
   c. Bandingkan Nilai *Cheksum* Semua File
      Tahapan ini merupakan tahapan proses membandingkan nilai *checksum* file pertama dibaca dengan nilai *checksum* file yang lain, jika nilai *cheksum*nya sama maka file tersebut dinyatakan sama atau duplikat
4. *Design*
   Tahapan ini merupakan tahapan peracangan dalam membangun sistem yang sesuai dengan domain masalah yang dikaji. Adapun perancangan yang dilakukan sesuai dengan hasil analisis dalam melakukan pencarian dan penghapusan file duplikat dengan algoritma CRC32 sebagai berikut :
   a. Rancangan Algoritma



Gambar 3. 3 Flow chart Pencarian dan Penghapusan File Duplikat

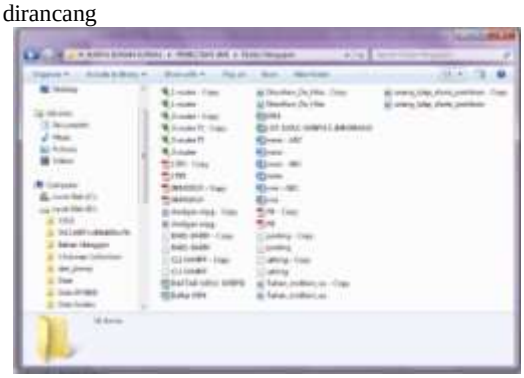
5. *Coding*
   Mentransformasikan algoritma dari hasil analisa yang di dapat terhadap sistem yang didesain dan melakukan pengujian secara *partial* dengan menggunakan salah satu bahasa pemrograman untuk dijadikan sebagai sistem pemrosesan untuk mencari dan menghapus file yang duplikat pada media peyimpanan komputer.

6. *Testing / Evaluation*
   Merupakan langkah yang digunakan untuk menguji, dengan berbagai bentuk unjuk kerja, untuk melihat perilaku sistem, hingga melakukan validasi dan menganalisis hasil akhir (*output*) yang diperoleh dari sistem yang dirancang serta melakukan evaluasi ketepatan terhadap kinerja sistem untuk menarik kesimpulan sesuai dengan domain masalah yang dikaji.

4. Hasil dan Pembahasan

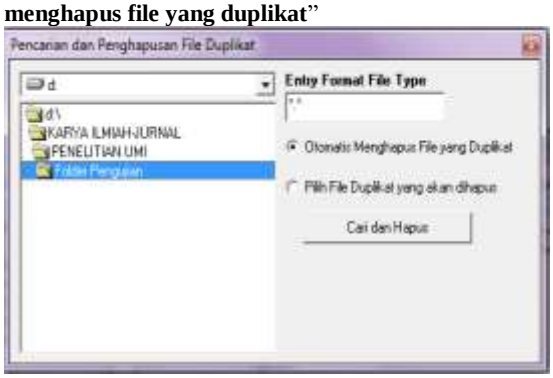
Luaran dari sistem pencarian dan penghapusan file duplikat dengan algoritma CRC 32 sebagai berikut :

1. Tampilan file manager pada windows explorer pada Folder yang dirujuk untuk mendeteksi file yang duplikat sebelum dihapus dengan system yang dirancang



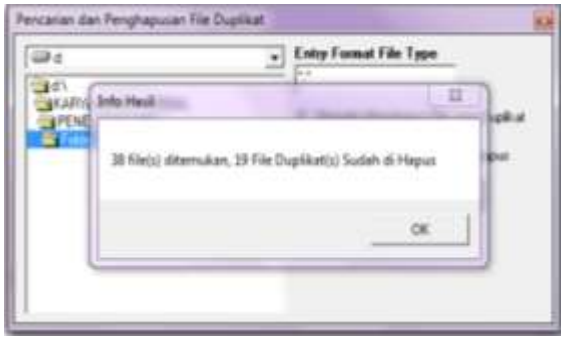
Gambar 4.1 Tampilan File Manager Windows Explorer sebelum di hapus

2. Proses Pencarian file duplikat dengan pilihan "**Otomatis menghapus file yang duplikat**"



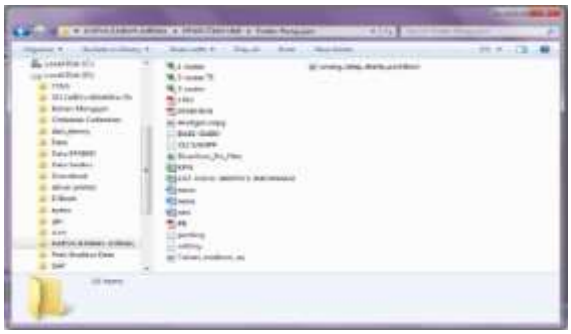
Gambar 4.2 Proses Pencarian dan Pengahapusan File Duplikat secara otomatis

3. Luaran Pecarian dan Penghapusan file duplikat secara otomatis



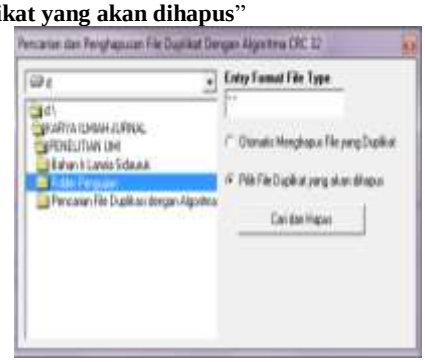
Gambar 4.3 Luaran Pencarian dan Penghapusan file duplikat secara otomatis

4. Tampilan file manager pada windows explorer pada Folder yang dirujuk setelah diproses dengan system yang dirancang



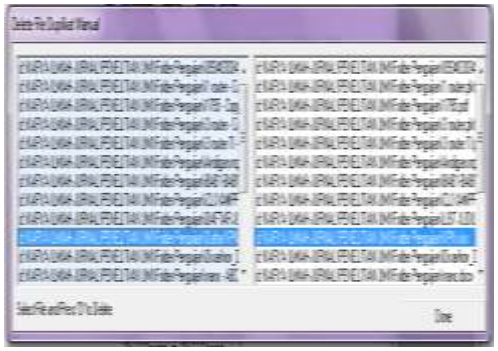
Gambar 4.4 Tampilan File Manager Windows Explorer setelah di hapus

5. Proses Pencarian file duplikat dengan pilihan "**Pilih File Duplikat yang akan dihapus**"



Gambar 4.5 Proses Pencarian dan Pengahapusan File Duplikat secara manual

6. Luaran Pecarian dan Penghapusan file duplikat secara manual



Gambar 4.6 Luaran Pencarian dan penghapusan file duplikat secara Manual

Hasil pengujian yang dilakukan untuk pencarian dan penghapusan file duplikat dengan algoritma CRC 32, dari beberapa koleksi file yang dikumpulkan dengan format file yang sama dan dengan nama file berbeda berhasil ditemukan. Adapun koleksi file yang digunakan dalam pengujian sistem ditunjukkan pada tabel-tabel di bawah

Tabel 4.1 Hasil Pengujian Pencarian dan Penghapusan File Duplikat Format File *.docx dan *.doc

No.	Nama File Asli	Nama File duplikasi	Format File	Ket.
1	nano	nano-ABC	.docx	Ditemukan
2	nene	nene-ABC	.docx	Ditemukan
5	Bab-1	Bab-1-Copy	.doc	Ditemukan

Tabel 4.2 Hasil Pengujian Pencarian dan Penghapusan File Duplikat Format File *.pdf

No	Nama File Asli	Nama File duplikasi	Format File	Ket
1	093403034	093403034 – Copy	.pdf	Ditemukan
2	1765	1765 – Copy	.pdf	Ditemukan
3	Jurnal Deteksi Virus dengan CRC32	Jurnal CRC-32	.pdf	Ditemukan

Tabel 4.3 Hasil Pengujian Pencarian dan Penghapusan File Duplikat Format File *.txt

No	Nama File Asli	Nama File duplikasi	Format File	Ket
1	Bab1-Babiv	Bab1-Babiv – Copy	.txt	Ditemukan
2	Cli Xampp	Cli Xampp - Copy	.txt	Ditemukan
3	Penting	penting - Copy	.txt	Ditemukan

Tabel 4.4 Hasil Pengujian Pencarian dan Penghapusan File Duplikat Format File *.mp4

No.	Nama File Asli	Nama File duplikasi	Format File	Ket
1	Andigan	Andigan - Copy	.mp4	Ditemukan
2	Disarihon_Do_Hita	Disarihon_Do_Hita - Copy	.mp4	Ditemukan
3	Tuhan_ondihon_au	Tuhan_ondihon_au - Copy	.mp4	Ditemukan

5. Kesimpulan

Berdasarkan analisis dan impelementasi sistem yang dilakukan untuk mendeteksi dan menghapus file duplikat dengan algoritma CRC 32 ditarik kesimpulan sebagai berikut :

1. Pencarian dan penghapusan file duplikat berhasil dilakukan apabila format type file sama walaupun nama filenya berbeda.
2. Pencarian dan penghapusan file duplikat dari berbagai macam format file yang dikumpulkan berhasil ditemukan dan dihapus, sehingga dinyatakan bahwa CRC 32 dapat mendeteksi semua format file yang duplikat.
3. Waktu proses pencarian dan penghapusan file sangat berpengaruh terhadap banyaknya format file dan besarnya media penyimpianan yang dicari.
4. Perubahan *content* pada file, seperti pada file gambar dengan format file sama dengan resolusi yang berbeda, maka CRC 32 tidak menyatakan file tersebut tidak duplikat

6. Daftar Pustaka

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

[2] M Sarkis Indra, "Kajian CRC-32 Untuk Mendeteksi Perubahan Isi File Document" Proceding SNIKOM, APTIKOM, 2015

[3] Narapratama, "Perbandingan Performansi Algoritma Adler-32 dan CRC-32 pada Library Zlib Bandung: Institut Teknologi Bandung, 2006

[4] Wijayanto,"Penggunaan CRC32 Dalam Integritas Data", Bandung: Institut Teknologi Bandung, 2006

[5] Silbertschatz, Galvin and Gagne,"Operating System Concepts 8 th Edition" Jhon Wiley and Sons Inc,United statea of America, 2009