



# KILAT

JURNAL KAJIAN ILMU DAN TEKNOLOGI

Rakhmadi Irfansyah Putra

Rizqia Cahyaningtyas  
Awit Lela Sigi

Dian Hartanti

Abdul Haris; Nina Nirmaya

Riki Ruli A. Siregar  
Alwi Baraqbah

Yessy Asri

Marliana Sari

Inge Handriani

Herman Bedi Agtriadi

Indah Handayasari  
Aziz Maulana

Irma Wirantina Kustanrika

Mukhlis Akhadi

PERANCANGAN APLIKASI PENGEMBALIAN BERKAS TERHAPUS PADA NTFS

PERANCANGAN APLIKASI PENGELOLAAN RUMAH TANGGA  
LABORATORIUM KOMPUTER STT-PLN

MODEL CLUSTERING MENGGUNAKAN ALGORITMA K-MEANS PADA DATA  
KELUHAN PELANGGAN PT PLN (PERSERO)  
(STUDI KASUS : PT PLN (PERSERO) DISTRIBUSI JAKARTA DAN TANGERANG)

APLIKASI KONVERSI AKSARA SUNDA KE BAHASA INDONESIA BERBASIS WEB  
MENGGUNAKAN PHP MYSQL

SISTEM KONTROL PADA PENGGUNAAN ENERGI LISTRIK BERBASIS ANDROID  
STUDI KASUS : SEKOLAH MENENGAH PERTAMA NEGERI (SMPN) 7 PONTIANAK

ANALISA PERBANDINGAN KEPUTUSAN METODE KLASIFIKASI DECISION TREE  
DAN NAÏVE BAYES DALAM PENENTUAN DIAGNOSA HIPERTENSI

PUSAT INFORMASI KEMAHASISWAAN DENGAN MENGGUNAKAN PHP, MYSQL  
DAN METODE MVC

FLOWCHART SISTEM PENAGIHAN PADA PERUSAHAAN JASA KONSULTAN

RANCANG BANGUN SISTEM MONITORING PADA BANGUNAN PENGENDALI  
SEDIMEN BERBASIS ANDROID DI PLTA

DESAIN ALTERNATIF JEMBATAN MENGGUNAKAN PLAT GIRDER  
(STUDI KASUS JEMBATAN RSUD KOTA TANGERANG)

PERHITUNGAN SINYAL PADA SIMPANG DENGAN METODE WEBSTER

MEMPRODUKSI BAHAN SEMIKONDUKTOR DI DALAM TERAS REAKTOR NUKLIR

ISSN 2089-1245



9 772089 124519

SEKOLAH TINGGI TEKNIK - PLN (STT-PLN)

KILAT

VOL.4

NO.1

HAL. 1 - 119

APRIL 2015

ISSN 2089 - 1245

# PERANCANGAN APLIKASI PENGEMBALIAN BERKAS TERHAPUS PADA NTFS

Rakhmadi Irfansyah Putra., SKom., MMSI

Dosen Teknik Informatika STT-PLN rakhmadi\_i@sttpln.ac.id

Adith Aulia Rahman

Teknik Informatika STT-PLN me@adithmd.my.id

## Abstract

*NTFS is a high-performance and self healing file system proprietary to Windows XP Vista 2008 2003 2000 NT & Windows 7, Windows 8, which supports file-level security, compression and auditing. It also supports large volumes and powerful storage solution such as RAID. The most important features of NTFS are data integrity (transaction journal) and the ability to encrypt files and folders to protect your sensitive data.*

*The waterfall development model originates in the manufacturing and construction industries; highly structured physical environments in which after the fact changes are prohibitively costly, if not impossible. Since no formal software development methodologies existed at the time, this hardware-oriented model was simply adapted for software development.*

*File recovery process can be briefly described as drive or folder scanning to find deleted entries in Master File Table (MFT) then for the particular deleted entry, defining clusters chain to be recovered and then copying contents of these clusters to the newly created file.*

## Abstrak

*NTFS berkinerja tinggi dan penyembuhan eksklusif sistem file untuk Windows XP Vista 2008 2003 2000 NT & Windows 7, Windows 8, yang mendukung file-level keamanan, kompresi dan audit. Ini juga mendukung volume besar dan solusi penyimpanan yang kuat seperti RAID. Fitur yang paling penting dari NTFS adalah integritas data (jurnal transaksi) dan kemampuan untuk mengenkripsi file dan folder untuk melindungi data sensitif Anda.*

*Model pembangunan air terjun berasal dalam industri manufaktur dan konstruksi; lingkungan fisik sangat terstruktur di mana setelah fakta perubahan mahal, jika tidak mustahil. Karena tidak ada perangkat lunak formal pengembangan metodologi ada saat itu, model berorientasi perangkat keras ini disesuaikan untuk pengembangan perangkat lunak.*

*Proses pemulihan file dapat dijelaskan secara singkat sebagai pemindaian drive atau folder untuk menemukan entri dihapus dalam Master File Table (MFT) kemudian untuk entri dihapus tertentu, mendefinisikan rantai kluster untuk memulihkan dan kemudian menyalin isi dari kluster ini ke file yang baru dibuat.*

**Kata kunci :** *ntfs, waterfall methodology, ntfs recovery.*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Pengelolaan data pada sistem operasi sangatlah penting, dengan adanya manajemen sistem berkas membuat user jauh lebih mudah dalam mengelola data. Seperti yang telah kita ketahui, komputer dapat menyimpan informasi ke beberapa media penyimpanan yang berbeda, seperti *magnetic disks*, *magnetic tapes*, dan *optical disks*. Agar komputer dapat digunakan dengan nyaman, sistem operasi menyediakan sistem penyimpanan dengan sistematis yang seragam. Sistem Operasi mengabstraksi properti fisik dari media penyimpanannya dan mendefinisikan unit penyimpanan logis, yaitu berkas. Berkas dipetakan ke media fisik oleh sistem operasi. Berkas adalah kumpulan informasi berkait yang diberi nama dan direkam pada penyimpanan sekunder. Dari sudut pandang pengguna, berkas merupakan bagian terkecil dari penyimpanan logis, artinya data tidak dapat ditulis ke penyimpanan sekunder kecuali jika berada di dalam berkas. Biasanya berkas merepresentasikan program

(baik *source* mau pun bentuk objek) dan data. Data dari berkas dapat bersifat numerik, alfabetik, alfanumerik, atau pun biner. Format berkas juga bisa bebas, misalnya berkas teks, atau dapat juga diformat pasti. Secara umum, berkas adalah urutan bit, byte, baris, atau catatan yang didefinisikan oleh pembuat berkas dan pengguna. NTFS merupakan salah satu dari sekian banyak manajemen sistem berkas pada sistem operasi. NTFS (*New Technology File System*) merupakan sistem berkas yang diperkenalkan Microsoft pada lingkungan windows NT. Dengan windows sebagai penguasa pasar sistem operasi basis desktop dan pendukung NTFS, tentunya membuat pengguna membutuhkan manajemen sistem berkas andalan Microsoft tersebut. Kenyataan seringnya kehilangan data oleh faktor lain terkadang sering membuat beberapa pengguna frustrasi. Pada beberapa kasus pengguna sering kehilangan data karena tanpa sengaja terhapus, dan ketika diperiksa pada *Recycle bin* tidak ditemukan. Tentunya pengguna akan kesulitan apabila datanya hilang begitu saja.

Dalam sistem berkas NTFS, semua data yang tersimpan di dalam sebuah volume disimpan dalam bentuk berkas. Hal ini berlaku untuk direktori, berkas biasa, hingga struktur data yang dimiliki oleh NTFS itu sendiri (yang disebut dengan NTFS Metadata). Ketika volume diformat dengan NTFS, file Master File Table (MFT) dan bagian lain dari metadata diciptakan. Metadata adalah file NTFS digunakan untuk menerapkan struktur sistem file. NTFS menciptakan catatan file untuk setiap file dan catatan direktori untuk setiap direktori yang dibuat pada volume NTFS. MFT termasuk catatan file terpisah untuk MFT itu sendiri. File dan direktori catatan ini disimpan pada MFT. Atribut file ditulis ke ruang yang dialokasikan dalam MFT. Selain atribut berkas, masing-masing record file berisi informasi tentang posisi dari catatan file dalam MFT. Setiap file biasanya menggunakan satu file rekaman. Namun, jika file memiliki sejumlah besar atribut atau menjadi sangat terfragmentasi, mungkin membutuhkan lebih dari satu file rekaman. Jika hal ini terjadi, catatan pertama untuk file, yang disebut catatan file base, menyimpan lokasi catatan file lain yang dibutuhkan oleh file. File kecil dan direktori (biasanya 1.500 bytes atau lebih kecil) yang seluruhnya terkandung dalam catatan MFT file. Catatan Direktori berisi informasi indeks. Direktori kecil mungkin berada sepenuhnya dalam struktur MFT, sedangkan direktori besar diatur dalam struktur B-tree dan memiliki catatan dengan pointer ke cluster eksternal yang berisi entri direktori yang tidak bisa terkandung dalam struktur MFT. Dengan memanfaatkan Master File Table pada NTFS penulis akan melakukan penelitian berdasarkan masalah yang dihadapi pengguna dalam mengembalikan file yang hilang, meskipun penulis tidak menjamin file akan 100% kembali secara utuh.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka penulis akan memberikan rumusan masalah sebagai berikut :

1. Bagaimana implementasi pengembalian berkas pada file system NTFS.
2. Bagaimana implementasi pemindaian berkas terhapus pada NTFS, yang terkait pada *Master File Table*.
3. Bagaimana mencari info terkait dengan file terhapus agar lebih mempermudah pengguna aplikasi dalam pencarian data.
4. Bagaiman mendesain GUI (antarmuka) untuk memudahkan pengguna aplikasi dalam menjalankan aplikasi ?
5. Bagaimana membuat status persentase keberhasilan pengembalian berkas ?

## 1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Pengembalian berkas yang terhapus dan tidak berada pada *Recycle bin*.
2. Pengembalian berkas ini bukan kepada fisik atau harddisk yang rusak, melainkan *software* yang hilang pada *file system* didalam harddisk.
3. File system yang digunakan dalam pengembalian berkas hanyalah NTFS, dan bukan File System lainnya seperti EXT, FAT16, FAT32 & dll.

4. Pengembalian berkas hanya kepada file berkas yang terhapus, bukan terformat.
5. Pengambalian file bukan terhadap file yang ada di *Recycle bin* dan dilakukan pengosongan / penghapusan data pada *Recycle bin*.
6. Pengambalian berkas terhapus dapat dilakukan dengan beberapa kondisi tertentu, dan tidak secara pasti dapat mengembalikan berkas terhapus kembali ke kondisi terakhir berkas terhapus.

## 1.4 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Membantu pengguna mengatasi kesulitan dalam mengembalikan berkas yang terhapus.
2. Menghasilkan aplikasi Pengembalian berkas yang tepat dan efisien.
3. Mempelajari metode dan cara kerja pada file system NTFS.

## 1.5 Manfaat

Manfaat dari penelitian ini adalah:

1. Memberikan kemudahan terhadap pengguna dalam melakukan penembalian data.
2. Dapat mengetahui status file yang terhapus dapat dikembalikan dengan persentase keberhasilan.
3. Dengan mudah dapat menemukan file yang dibutuhkan.
4. Mengurangi resiko kehilangan file.

## 2. LANDASAN TEORI

### 2.1 Sistem Berkas

Sebuah sistem manajemen file adalah seperangkat sistem perangkat lunak yang memberikan layanan kepada pengguna dan aplikasi dalam penggunaan file. Biasanya, satu-satunya cara bahwa pengguna atau aplikasi dapat mengakses file adalah melalui sistem manajemen file. Hal ini mengurangi pengguna atau programmer perlunya mengembangkan perangkat lunak tujuan khusus untuk masing-masing aplikasi dan menyediakan sistem dengan konsisten, berarti didefinisikan dengan baik mengendalikan aset yang paling penting [1]. *A file system is a part of the operating system on a volume and it determines how files are named, stored, and organized on basic or dynamic disks. A file system manages files and folders as well as the information required to locate and access these items by local and remote users* [2]. Semua sistem operasi mulai dari DOS, Windows, Macintosh dan turunan UNIX memiliki Sistem berkas sendiri untuk meletakkan file dalam sebuah struktur hirarki. Contoh dari sistem berkas termasuk di dalamnya FAT, NTFS, HFS dan HFS+, ext2, ext3, ISO 9660, ODS-5, dan UDF

### 2.2 NTFS

*NTFS provides performance, reliability, and advanced features not found in FAT. For example, NTFS includes built-in security features such as file and folder permissions and the Encrypting File System (EFS), which is the technology used to store encrypted files on NTFS volumes. NTFS also provides support for volumes up to 256 terabytes in size, support for disk*

quotas and compression, and support for mounted drives[2]. Sebuah sistem berkas merupakan bagian dari sistem operasi pada volume dan menentukan bagaimana file diberi nama, disimpan, dan terorganisir pada disk dasar atau dinamis. Sebuah sistem file mengatur file dan *folder* serta informasi yang dibutuhkan untuk mencari dan mengakses item ini oleh pengguna lokal dan jarak jauh. NTFS menyediakan performa, kehandalan, dan fitur-fitur canggih yang tidak ditemukan di FAT. Sebagai contoh, NTFS termasuk built-in fitur keamanan seperti file dan hak akses folder dan *Encrypting File System* (EFS), yang merupakan teknologi yang digunakan untuk menyimpan file terenkripsi pada volume NTFS. NTFS juga menyediakan dukungan untuk volume hingga 256 terabyte dalam ukuran, dukungan untuk kuota disk dan kompresi, dan dukungan untuk *drive* terpasang.

### 2.3 Pengembalian Berkas

Proses pemulihan file dapat dijelaskan secara singkat sebagai pemindaian *drive* atau *folder* untuk menemukan entri dihapus dalam Master File Table (MFT) kemudian untuk entri tertentu yang terhapus, mendefinisikan cluster rantai untuk dipulihkan dan kemudian menyalin isi dari kelompok ini ke file yang baru dibuat [3]. NTFS menggunakan transaksi *logging* dan pemulihan untuk menjamin bahwa struktur volume tidak rusak. Untuk alasan ini, semua file sistem tetap dapat diakses setelah kegagalan sistem. Namun, data pengguna dapat hilang karena kegagalan sistem atau *bad sector*[10]. Sistem file yang berbeda mempertahankan struktur spesifik data logis mereka sendiri, namun pada dasarnya setiap sistem file[4]:

- Memiliki daftar atau katalog *entri file*, sehingga kita dapat *iterate* melalui daftar ini dan entri, ditandai sebagai dihapus
- Terus untuk setiap entri daftar *cluster* data, sehingga kita dapat mencoba untuk mencari tahu *set cluster* menyusun file

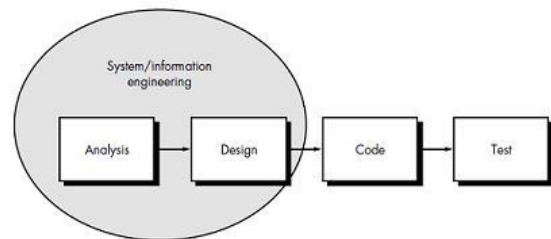
Setelah mengetahui entri file yang tepat dan perakitan *set cluster*, menyusun file, membaca dan menyalin kelompok ini ke lokasi lain. Namun, tidak setiap file yang dihapus dapat dipulihkan, ada beberapa asumsi, pasti:

- Pertama, kita mengasumsikan bahwa entri file masih ada (tidak ditimpa dengan data lain). Semakin sedikit file yang telah dibuat pada *drive* di mana file yang dihapus itu tinggal, semakin banyak kesempatan bahwa ruang untuk entri file yang dihapus tidak digunakan untuk entri lainnya.
- Kedua, kita asumsikan bahwa entri file lebih atau kurang aman untuk menunjuk ke tempat yang tepat di mana cluster berkas berada. Dalam beberapa kasus (telah menyadari pada Windows XP, pada volume FAT32 besar) sistem operasi kerusakan mengajukan masukan tepat setelah penghapusan sehingga *cluster* data yang pertama menjadi tidak berlaku dan restorasi entri lebih lanjut tidak mungkin.
- Ketiga, kita asumsikan bahwa *cluster* data file aman (tidak ditimpa dengan data lain). Semakin sedikit operasi *write* telah dilakukan pada *drive* dimana file yang dihapus itu tinggal, semakin banyak kesempatan bahwa ruang yang ditempati oleh kelompok data file yang dihapus tidak digunakan untuk penyimpanan data lainnya

## 3. Metodologi Penelitian

### 3.1 Kerangka Pemikiran

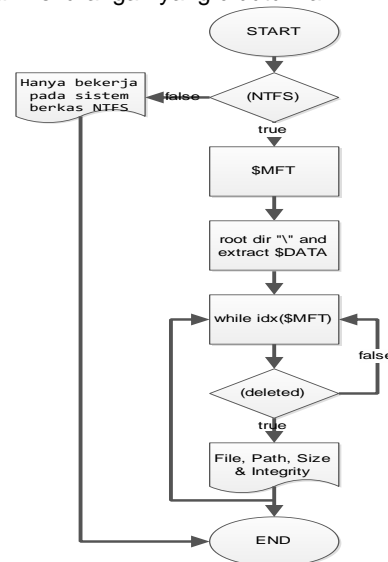
Secara umum, pemikiran dalam perancangan aplikasi ini tersaji dalam diagram alir kerangka pemikiran di bawah ini yang menggunakan Metode *Waterfall*. Diagram tersebut memperlihatkan tahap-tahap proses penelitian. Terdapat 4 tahapan langkah dalam menggunakan metode ini dalam pengembalian data terhapus.



Gambar 1 Waterfall diagram

Metode *waterfall* digunakan dalam membangun kerangka sistem aplikasi berdasarkan kebutuhan.

- Analisis diperlukan untuk mengetahui kebutuhan yang diperlukan dalam perancangan sistem.
- Design* dirancang sesuai berdasarkan hasil dari analisis kebutuhan, dan sebisa mungkin dibuat semudah mungkin untuk memudahkan pengguna.
- Pada tahapan *code*, dibuat untuk memperkuat kerangka sistem dalam pembuatan aplikasi
- Setelah ketiga tahapan sebelumnya selesai, maka dilakukan test untuk menganalisa *bug track* dan kekurangan yang dibutuhkan.



Gambar 2 Flowchart Prototype

- Menemukan entri partisi yang mempunyai partisi NTFS
- Arahkan ke awal Master File Table
- Temukan metafile entry direktori root di MFT dan ekstrak indeks Alokasi data atribut.
- Proses catatan INDX ditemukan dalam data indeks alokasi atribut satu per satu rekursif

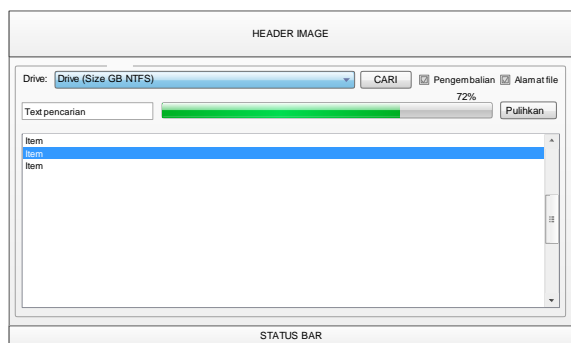
- sampai Anda menemukan file yang cocok dengan satu Anda cari.
5. Dalam entri indeks yang cocok, menemukan nomor catatan MFT dan pindah ke posisi record dalam MFT.
  6. Catat entri MFT dan memproses atribut *header* standar sampai atribut data yang ditemui.
  7. Gunakan proses ekstraksi data atribut dalam rangka untuk mengambil data atribut, yang berisi isi dari file yang sedang diakses.

### 3.2 Perancangan Sistem

Berdasarkan analisis permasalahan yang ada, maka pada tahap ini dijelaskan rancangan pengembalian berkas terhapus pada NTFS. Poses perancangan pengembangan ini akan digambarkan menggunakan Flowchart.

#### 3.2.1 Antarmuka

Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.



Gambar 3 Antarmuka Sistem

Pada gambar 3.6 tampak penampakan antarmuka dari hasil perancangan aplikasi pengembalian sistem berkas pada NTFS. Antarmuka dibuat secara sederhana agar memudahkan pengguna dalam menggunakan aplikasi. Pengguna hanya melakukan beberapa interaksi untuk menjalankan alur pengembalian file, sehingga apabila pengguna awam akan teknologi tidaklah terlalu sulit untuk mengoperasikannya.

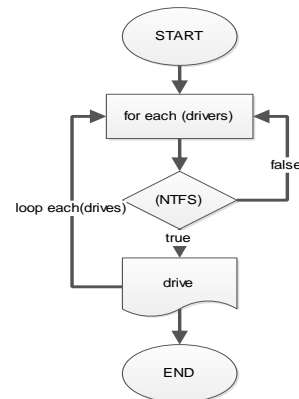
#### 3.2.2 Flowchart

Alur flowchart dibutuhkan dalam pemahaman kebutuhan data dan proses dalam *recovery file*. Berikut merupakan beberapa alur *flowchart* dalam antarmuka untuk menampilkan alur rancang bangun aplikasi.

Adapun diagram flowchart yang tertera adalah:

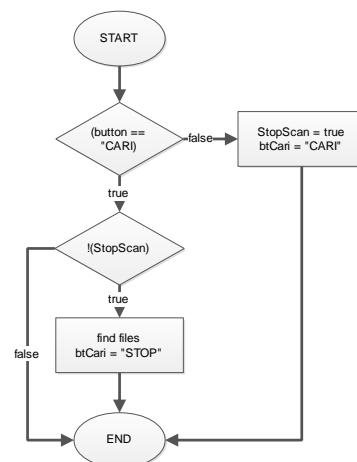
- a. Flowchart file system
- b. Flowchart sistem proses

- c. Flowchart file scan/stop button
- d. Flowchart file finding



Gambar 4 Flowchart File System

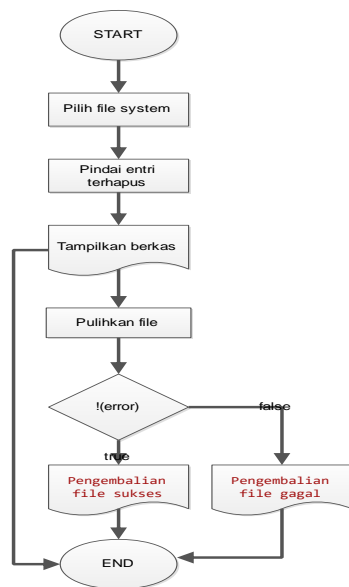
Pada Gambar 3.3 Dari alur flowchart diatas akan menampilkan output berupa susunan file system NTFS yang terdapat pada harddisk. Buka disk untuk membaca dan menulis raw data adalah yang sederhana seperti membuka perangkat `\\.\DriveN` dimana N merupakan jumlah drive yang Anda inginkan untuk dibuka untuk membaca dan menulis raw data. Beberapa langkah harus dilakukan untuk detail langkah-langkah yang diperlukan untuk mengidentifikasi jenis partisi lain pada disk fisik sehingga kita dapat mengidentifikasi partisi adalah benar-benar NTFS. Setelah kita mendapatkan disk dengan partisi sistem valid NTFS, maka simpan / tampilkan drive tersebut kedalam combo box.



Gambar 5 Flowchart File Scan/Stop button events

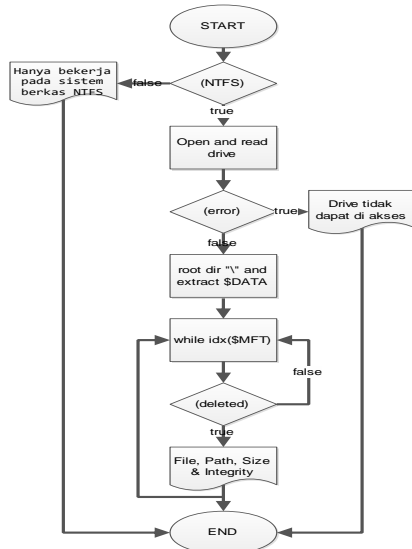
Bagan flowchart menampilkan alur bagaimana sistem kerja tombol "CARI" melaksanakan fungsi pencarian dan melaksanakan fungsi "STOP" untuk melakukan penghentian pencarian. Apabila pengguna melakukan perintah pada saat tombol "CARI", maka *event* akan memerintahkan untuk memindai file berkas terhapus, dan nama pada tombol dirubah menjadi "STOP". Apabila pengguna melakukan perintah pada saat tombol "STOP", maka *event* akan memerintahkan pemberhentian *event* pencarian file berkas terhapus dan mengubah kembali tombol dengan nama "CARI".





**Gambar 6** Flowchart sistem proses

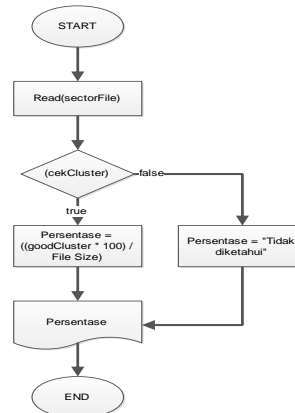
Pada Gambar diatas Bagan flowchart diatas akan menampilkan entri dari berkas yang terhapus sebelumnya dengan memilih file system yang akan diperiksa. Bagan ini merupakan rangkaian sistem sederhana dalam melakukan pencarian entri berkas dari data terhapus.



**Gambar 7** Flowchart File finding

Pada gambar diatas merupakan bagan flowchart dari pencarian file berkas terhapus pada NTFS. Ketika pengguna telah melakukan event "CARI", maka event selanjutnya yang akan dijalankan terdapat pada bagan flowchart berikut. Pada tahapan awal sistem akan membaca file system yang dipilih apakah benar valid NTFS? Bila bukan NTFS, maka akan menampilkan pesan "Hanya bekerja pada sistem berkas NTFS" dan langsung menghentikan event pencarian. Selanjutnya sistem akan membaca isi disk dan akan menuju pada \$MFT yang terdapat pada NTFS. Setelah

mendapatkan data didalam \$MFT, maka mulailah pemindaian entri yang terhapus pada NTFS dengan EOF pada \$MFT sebagai batasan akhir dari pencarian file terkait. Program akan mencari keseluruhan file pada metadata MFT, dan semua file yang terkait dengan MFT akan muncul meskipun file tanpa ekstensi bahkan file cache yang mempunyai ciri "~" pada awal file.

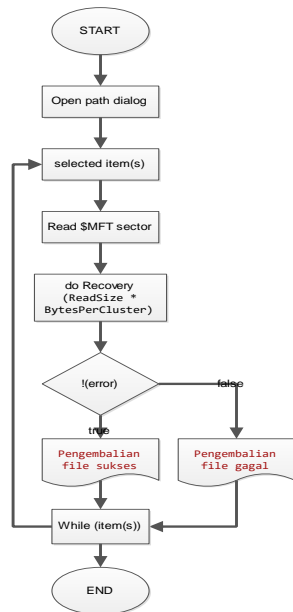


**Gambar 8** Flowchart Persentase file

Untuk dapat mengetahui persentase keberhasilan file dalam melakukan pengembalian data, harus dilakukan kalkulasi dengan cara sebagaimana bagan flowchart diatas. Pada flowchart diatas, pada saat melakukan pencarian index file di \$MFT, maka dilakukan pula pencarian persentase pengembalian data dengan membuka ke sector file. Setelah menuju file sector, maka akan dilakukan pengecekan cluster dengan mengacu pada data \$BITMAP yang ada pada \$MFT, apakah file berkas di cluster tersebut masih bagus (dapat dikembalikan), atau sudah terisi dengan file yang baru dibuat dan menempati cluster tersebut. Bila telah melakukan pengecekan cluster, tahap selanjutnya adalah melakukan kalkulasi persentase keberhasilan pengembalian file. Dengan rumusan :

$$\text{Percent} = (\text{GoodClusters} * 100) \setminus \text{FileSize}$$

Berdasarkan pustaka bahwa NTFS dapat melakukan pengembalian data jika entri tidak pada *bad cluster* atau karena kegagalan sistem. Jadi disimpulkan *good cluster* dan *bad sector* pada disk mempengaruhi persentase keberhasilan pengembalian data.



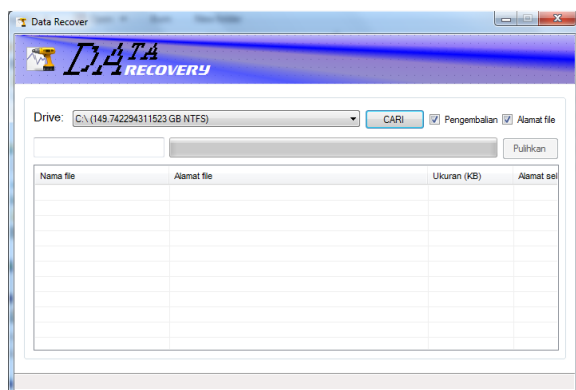
**Gambar 9** Flowchart Recovery file

Untuk melakukan pengembalian file, pilih terlebih dahulu file yang akan dikembalikan. Lalu pilih tempat dimana file akan dikembalikan. Maka program akan membaca *file sector (offset)* dari file yang dipilih dan membaca ke sector \$MFT sector file tersebut. Setelah menuju tempat file offset lakukan pengembalian baca dengan membaca good cluster dari offset file dan lakukan ekstrasi dengan melakukan ekstrasi byte dari file (Size \* bytesPerCluster). Selama melakukan pengembalian file / ekstraksi data program akan melakukan cek, apabila terdapat error, maka pengembalian file gagal, jika tidak maka berhasil.

#### 4. Hasil dan Uji coba

Penulis akan memaparkan mengenai hasil dan pembahasan implementasi system yaitu tampilan interface pada sistem. Tahapan dalam melakukan pengembalian berkas terbagi dalam beberapa tahap, yaitu :

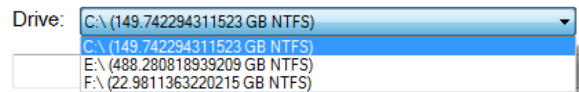
##### 1. Antarmuka



**Gambar 10** Antarmuka

Gambar diatas antarmuka awal ketika menjalankan aplikasi Data Recover.

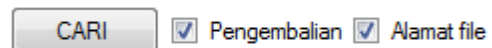
##### 2. Dropdown list drive



**Gambar 11** Dropdown list drive

Dropdown list drive merupakan daftar drive yang dapat dikenali oleh DataRecover. Pilih salah satu drive untuk dipindai.

##### 3. “Cari” button dan checklist



**Gambar 12** “Cari” button dan checklist

Gambar diatas adalah opsi selanjutnya ketika akan melakukan pemindaian drive. Klik tombol “CARI” untuk melakukan pemindaian. Cek “Pengembalian”, untuk mengetahui persentase keberhasilan pengembalian file. Cek “Alamat file”, untuk mengetahui alamat file.

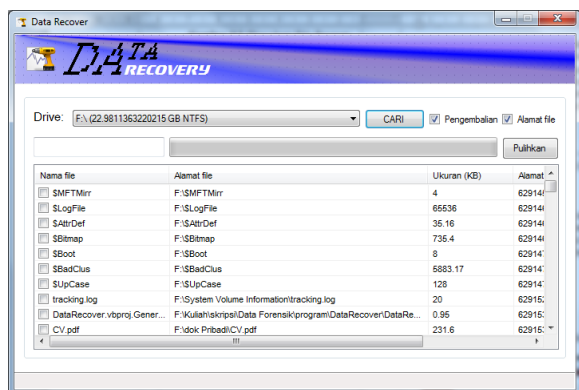
##### 4. Search progress



**Gambar 13** Search progress

Gambar diatas adalah progress saat melakukan pemindaian file, progress akan berhenti apabila progress bar telah sampai maksimal atau bisa klik “STOP” untuk memberhentikan pemindaian file.

##### 5. Recovery file



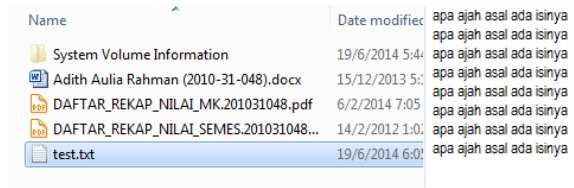
**Gambar 14** Recovery file

Gambar diatas adalah progress apabila pemindaian telah selesai. Pilih file yang akan dikembalikan, lalu klik “Pulihkan” untuk mengembalikan file.

##### Uji coba

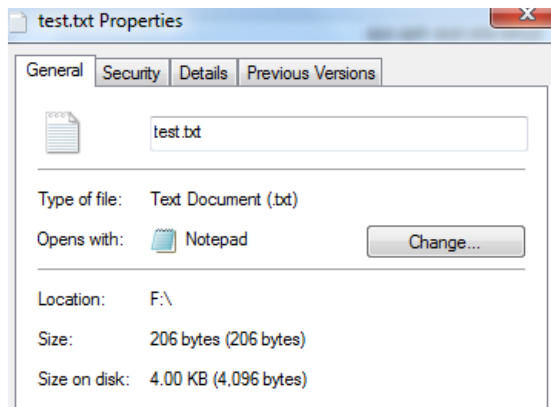
Untuk mencoba berhasil atau tidaknya aplikasi ini dalam mengembalikan berkas terhapus maka akan

dilakukan uji coba. Berikut screen shoot dan penjelasan dari hasil langkah demi langkah uji coba:



Gambar 15 Delete file

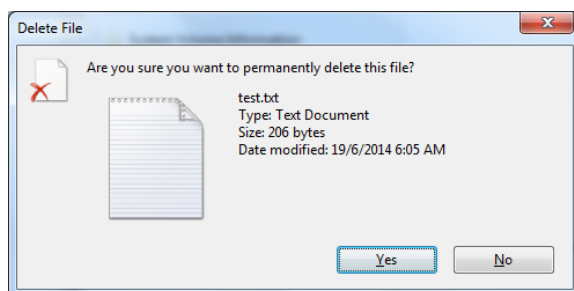
Tampak pada gambar diatas penulis melakukan uji coba penelitian penghapusan berkas “test.txt” dengan isi di dalamnya berupa “apa ajah asal ada isinya” sebanyak 8 baris.



Gambar 16 Detail file “test.txt”

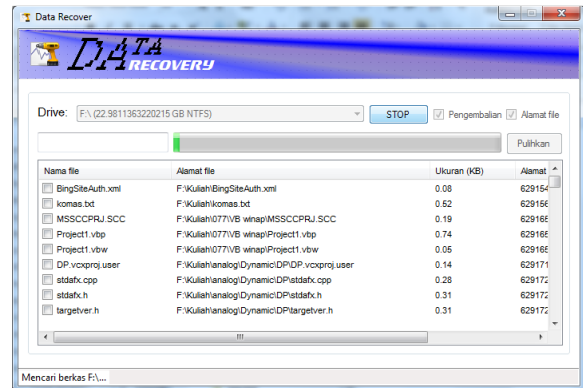
Pada gambar 4.7 terlihat jelas detail info dari file “test.txt” dengan detail sebagai berikut:

- o Lokasi : F\
- o Size : 206 bytes (0.2 KB)



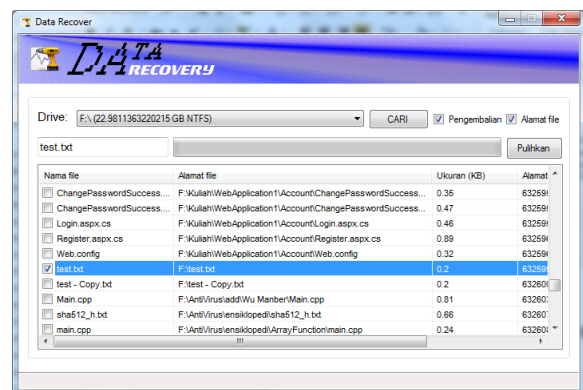
Gambar 17 Hapus file “test.txt”

Hapus file test.txt !



Gambar 18 Pindai file “test.txt”

Lalu lakukan pencarian file “test.txt”, tunggu hingga proses selesai berjalan.



Gambar 19 Cari file “test.txt”

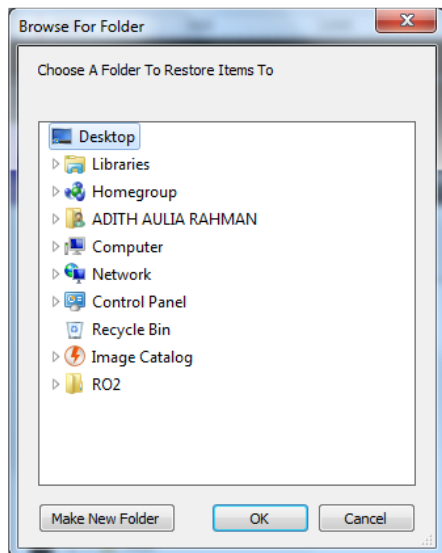
Apabila pencarian file telah berakhir, anda bisa mencari file “test.txt” dengan cara memasukkan kata “test.txt” pada textbox di sebelah kiri progressBar, maka file “test.txt” akan terpilih dan tercentang secara otomatis.

Alamat file	Ukuran (KB)	Alamat sekt...	Pengembalian
F:\Kuliah\WebApplication1\Account\Web.config	0.32	6325968	100 %
F:\test.txt	0.2	6325996	100 %

Gambar 20 Integritas pengembalian

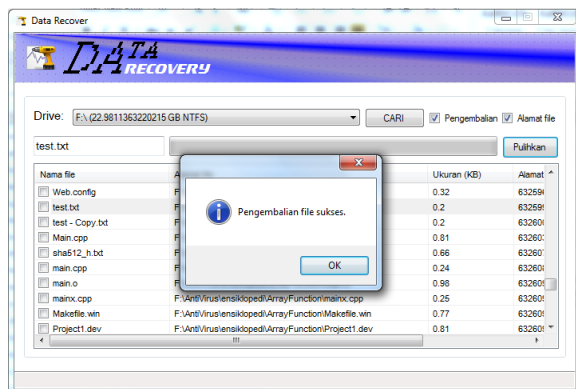
Jangan lupa untuk melihat integritas pengembalian, apakah file bisa dikembalikan secara seutuhnya ?





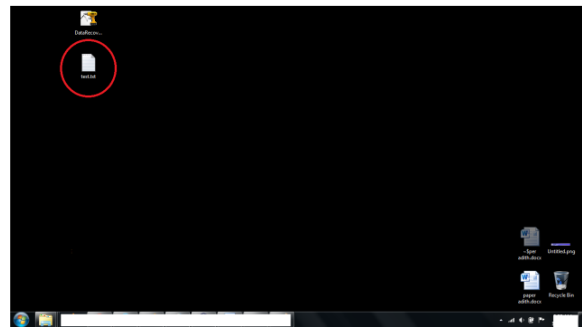
**Gambar 21** Pengembalian file

Setelah melakukan pengecekan integritas lakukan restorasi / pengembalian dengan cara melakukan klik pada tombol “Pulihkan”, lalu pilih alamat dimana file akan disimpan setelah melakukan pemulihan. Disini penulis akan memilih “Desktop” sebagai alamat pemulihan, agar lebih mudah.



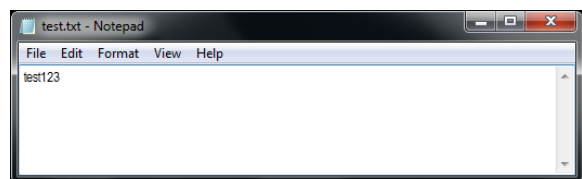
**Gambar 22** Info pemulihan

Setelah memilih alamat pengembalian file akan muncul info pesan “Pengembalian file sukses” apabila file telah berhasil dikembalikan ke “Desktop” alamat tempat penulis memilih tadi. Namun apabila gagal, maka akan muncul info pesan “Pengembalian file gagal”.



**Gambar 23** File pulih

Tampak pada layar desktop penulis bahwa file “Test.txt” (yang dilingkari) telah berhasil disimpan ke alamat “Desktop”. Namun untuk membuktikan bahwa file telah kembali secara utuh harus dilakukan pengecekan.



**Gambar 24** File pulih kembali

Tampak bahwa file telah kembali sesuai dengan file asli sebelumnya. Tentu hal ini bisa saja gagal, dan tidak kembali sepenuhnya dikarenakan hal-hal yang telah dijelaskan sebelumnya. Penulis sekali lagi tidak bisa menjamin 100% pengembalian secara utuh, dikarenakan keterbatasan dalam penelitian dan pengujian pada sistem NTFS.

## 5. Kesimpulan

Berdasarkan uraian-uraian yang telah dipaparkan, maka penulis menarik kesimpulan sebagai berikut:

1. Aplikasi pengembalian file system NTFS ini akan membantu pengguna untuk melakukan pengembalian berkas yang terhapus.
2. Setiap file yang ada pada file system NTFS berkaitan dengan Master File Table, dan melalui MFT dilakukan pencarian entri terhapus.
3. Pada MFT terdapat informasi mengenai file index yang telah terhapus, sehingga untuk mempermudah dalam mengenali file penulis dapat menampilkan informasi yang terdapat pada file sebelumnya di MFT.
4. Antarmuka dibuat semudah mungkin agar memudahkan pengguna dalam melakukan penggunaan aplikasi.
5. Status persentase file dihitung dengan memperhitungkan  $((\text{good cluster} * 100) / \text{besar file})$  untuk mengetahui persentase keberhasilan. Namun dapat pula gagal dikarenakan beberapa batasan yang terjadi.

## Daftar Pustaka

1. William Stallings, Operating System: Chapter 12.
2. NTFS, [http://technet.microsoft.com/en-us/library/cc778410\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc778410(v=ws.10).aspx) [8 November 2013].
3. NTFS Recovery, <http://ntfs.com> [8 November 2013].
4. Recovery concepts, <http://ntfs.com/file-recovery-concepts.htm> [8 November 2013].
5. John Medeiros, Exploring NTFS.
6. Visual Studio, <http://visualstudio.com>.
7. NET Framework, <http://www.microsoft.com/net>.
8. EdrawMax, <http://www.edrawsoft.com/download/edrawmanual.pdf> [8 Juni 2014].
9. NTFS Forensic, <http://grayscale-research.org/new/pdfs/NTFS%20forensics.pdf> [8 Juni 2014]
10. Recovering Data with NTFS, <http://technet.microsoft.com/en-us/library/cc976815.aspx> [26 Juni 2014].