LAB 6 ACTIVITIES

1. [MANAGE PARTITION] You are to create at least **THREE (3) partitions** in your USB storage with minimum of 40MB size.
2. [MANAGE PARTITION] Each partition should be formatted with **FAT**, **FAT32** and **NTFS**.
3. [WINDOW EXPLORER] **Copy PDF and JPG files** into these 3 partitions.
4. [FTK Imager] **Image** these partitions for each of the partition 2 times, 1 **BEFORE deletion** (e.g. named is as "B4-40MB-FAT-DEL" for FAT image) and 1 **AFTER deletion** (e.g. named is as "AFTER-40MB-FAT-DEL" for FAT image) using **FTK imager**.
5. [FTK Imager] Add Evidence Item in FTK Imager
    a. File -> **Add Evidence Item** -> "Please select source evidence type" -> Image file -> Choose image file (refer to **FIGURE 1** to **FIGURE 6**).
    b. Choose one of the deleted files (in the Root) and get the **properties** of a few deleted images e.g. **Start cluster**, **Start sector**, **File size**. (NOTE: **1 sector = 512 Bytes**, 1 cluster = 2 x sectors = 1024 Bytes)

    Assume that these information is obtained:
    File Size = 33,808
    Start Cluster = 10
    Start Sector = 16,444

    Calculating the **offset** of the beginning position of the deleted JPG using Start Sector:
    If the **Start Sector = 16,444**, it means that the starting position (offset) of the deleted image is from **16,444 x 512 (sector) = 8,419,328**.

    Calculating the **offset** of the end position of the deleted JPG using offset from the calculation above (8, 419, 328).

    8, 419,328 Offset of the starting position of deleted JPG
    + 33,808 File Size

    8, 453,136 Offset of the end position of deleted JPG
    ==========
6. Using this information, use the codes at the appendix and then recover the deleted file.

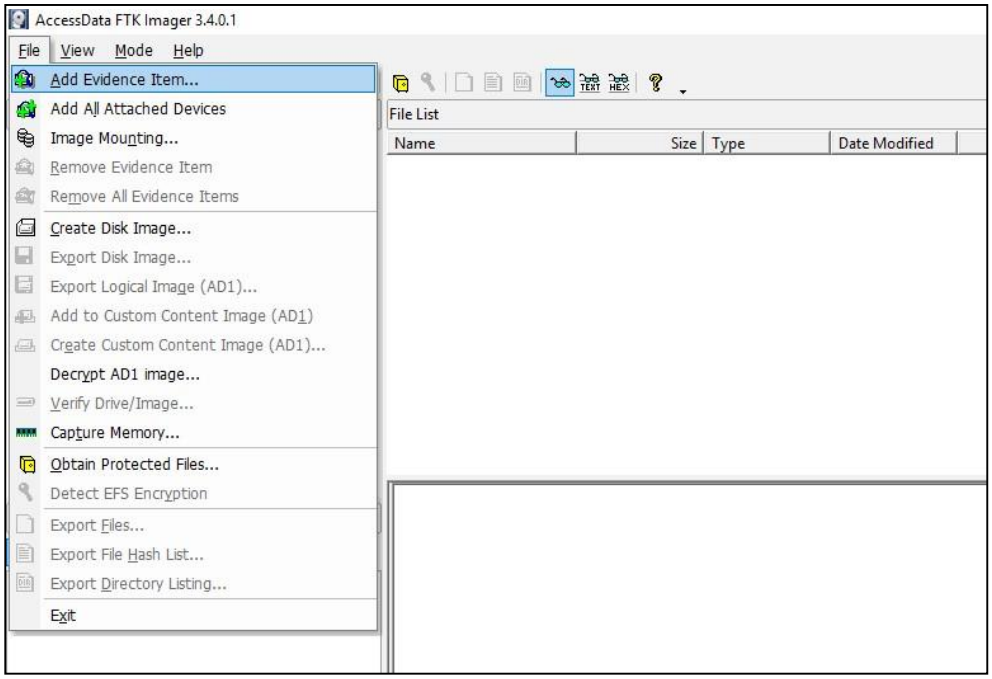

FIGURE 1: "**Add Evidence Item**" in FTK Imager

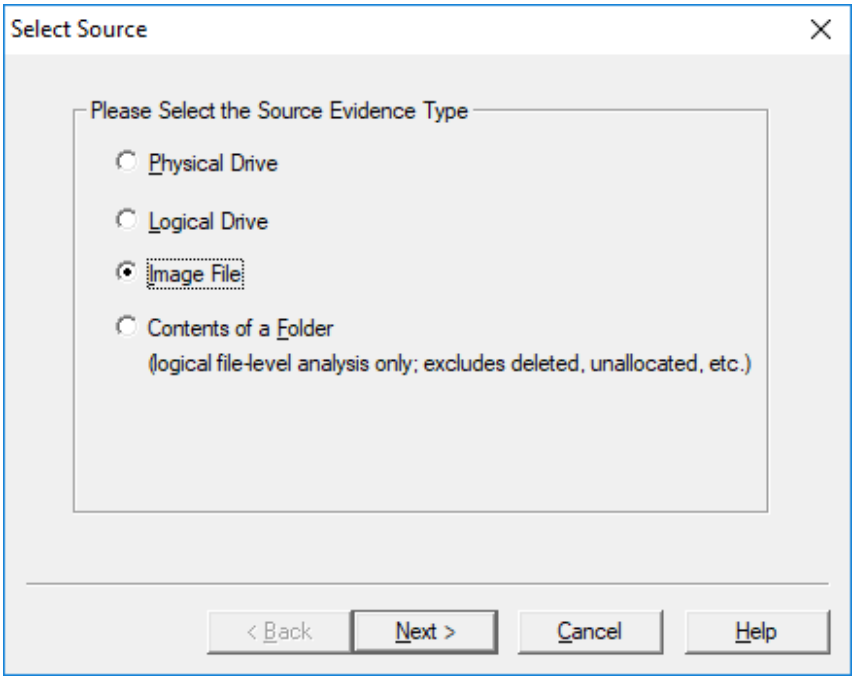

FIGURE 2: Select source from **Image File**.

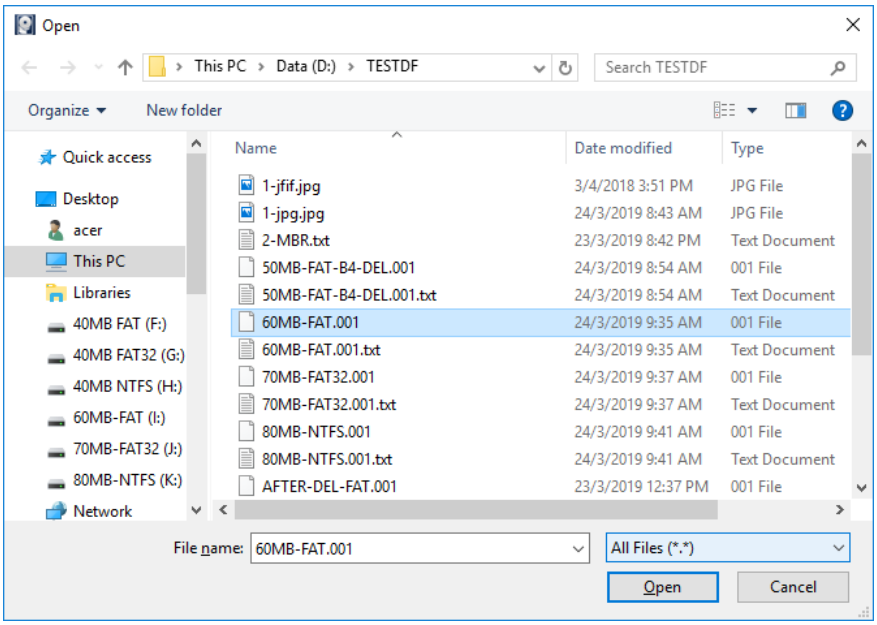

FIGURE 3: Browse for a raw (dd) image e.g. 60MB-FAT.001

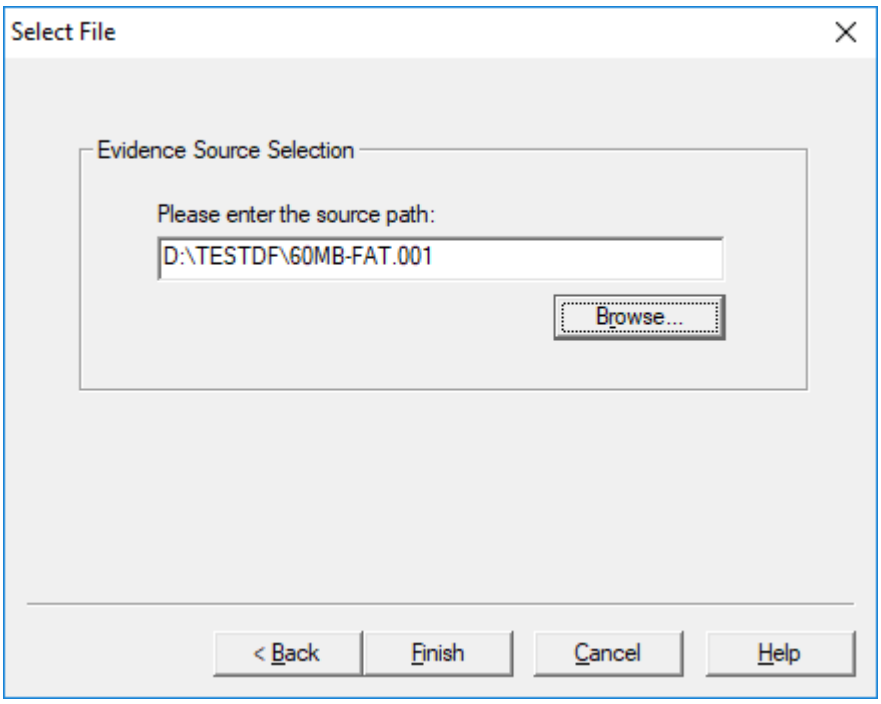

FIGURE 4: Raw (dd) image is selected e.g. 60MB-FAT.001

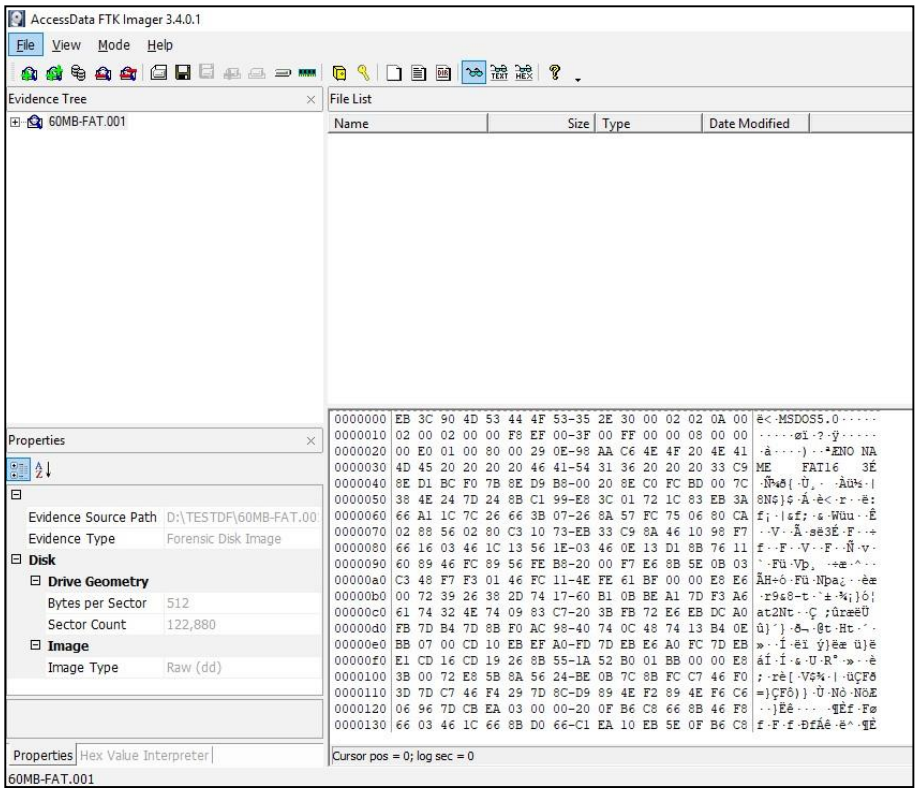

FIGURE 5: The image is opened as evidence item in FTK Imager

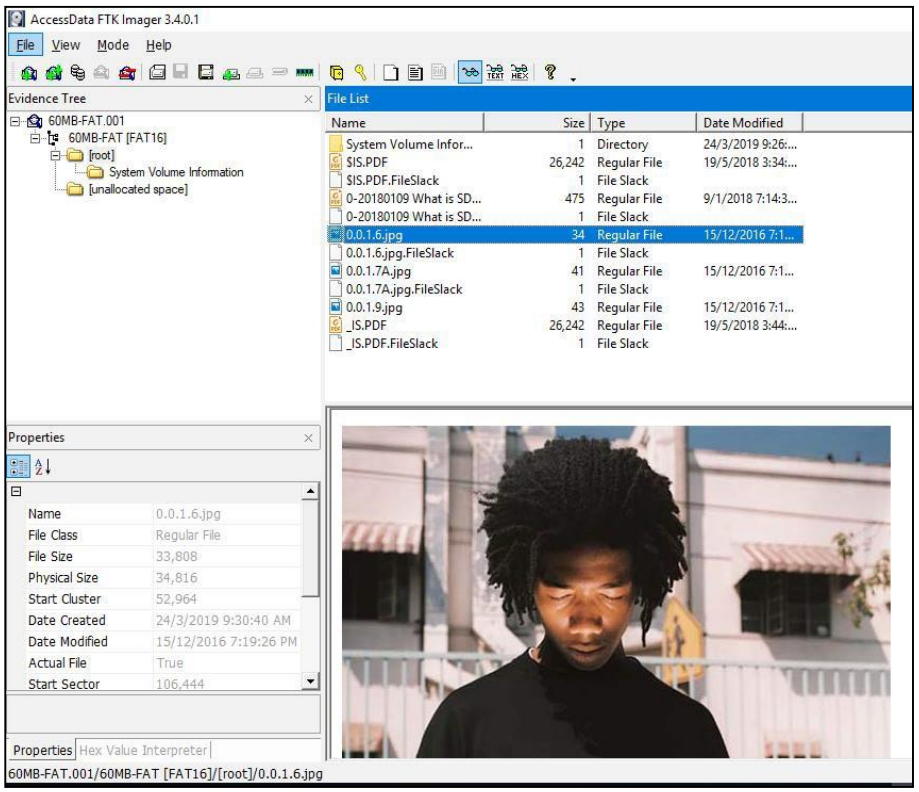

FIGURE 6: This is one of the deleted JPG file in the image which is able to be viewed using FTK Image

NOTE: When image is created using **FTK Imager**, you can change the default size 1500MB (or 1.5GB) of the image to any size. If the default size is used, multiple image files will be created of size 1.5GB each if the size of the storage to be image is bigger. These multiple size images can be combined into a single image file by using **ForensicImager** tool. This tool also can convert .E01 (Encase format) to RAW (dd) format.

QUESTIONS FOR THE LAB

1. Provide screenshots of the lab 6 activities

2. If the deleted file has this information:

   File Size = 2050 bytes

   Start Cluster = 3

   Start Sector = 12

   a) Assume that 1 cluster has 4 sectors, how many clusters are used?
   b) How many sectors are used?
   c) How much slack space is available?

3. Discuss the pros and cons of SMALLER CLUSTER (e.g. FAT) vs. BIGGER CLUSTER (e.g. NTFS).

APPENDIX

```
1.  /*
2.  20190324
3.  This program is to extract deleted JPG files from IMAGE created from a USB storage after files
4.  deletion (e.g. AFTER-DEL-FAT.001). Then, find the relevant information of the deleted jpg file using
5.  e.g. FTK Imager (from FTK properties, obtain the File Size, Start Cluster, Start Sector of the deleted
6.  jpg). Use this info and execute the program.
7.  */
8.  #include <stdio.h>
9.  #include <string.h>
10. #include <conio.h>   // to use getch()
11. #include <stdlib.h>   // to use exit()
12. main()
13. {
14.         FILE*
15.         image;
16.         FILE* jpg1;
17.         char filename1[100], filename2[100]; char
18.         c;
19.         long offset1, offset2, filesize;
20.
21.         strcpy(filename1,"d:\\TESTDF\\AFTER-DEL-FAT.001"); // image after deletion
22.         strcpy(filename2,"d:\\TESTDF\\1-jpg.jpg");
23.         image=fopen(filename1, "rb");  // open file for reading
24.         jpg1=fopen(filename2, "wb");   // open file for writing if
25.         (image==NULL)
26.         {
27.                 printf("FAILED to open IMAGE");
28.                 getch();
29.                 exit(1);
30.         }
31.         if (jpg1==NULL)
32.         {
33.                 printf("FAILED to open IMAGE");
34.                 getch();
35.                 exit(1);
36.         }
37.         offset1=6218752L;  // the starting address deleted jpg
38.         filesize=40189L;         // the size of the deleted jpg
39.         offset2=offset1+filesize; // the end offset of deleted jpg
40.
41.         fseek(image,offset1,SEEK_SET); // jump to offset1 from BEGINNING of FILE while
42.         (offset1<offset2)
43.         {
                c=fgetc(image);
                fputc(c,jpg1);
                offset1++;
        }
        fclose(image);
        fclose(jpg1);
}
```