# Tutorial Image Forensics pada Cyber Crime Step by Step dengan Kali Linux

Image forensics (forensik citra digital) adalah cabang forensik digital yang berfokus pada analisis keaslian gambar dan deteksi manipulasi. Dalam era digital ini, image forensics menjadi sangat penting untuk investigasi cybercrime, terutama dengan mudahnya melakukan manipulasi foto menggunakan software editing.

## Persiapan Lingkungan Forensik

## Boot Kali Linux dalam Mode Forensik

Seperti pada audio forensics, langkah pertama adalah boot Kali Linux dalam mode forensik untuk menjaga integritas bukti digital:[1]

```
# Boot dari Kali Live USB/CD
# Pilih "Forensic Mode" pada boot menu
```

**Karakteristik Mode Forensik:**

- Tidak me-mount hard disk internal secara otomatis
- Menonaktifkan swap partition
- Semua operasi pada media external harus dilakukan manual
- Mencegah perubahan tidak disengaja pada bukti digital

## Instalasi Tools Image Forensics

Kali Linux sudah dilengkapi dengan banyak tools forensik, namun beberapa perlu diinstal tambahan:[2][3]

```
# Update repository
sudo apt-get update && sudo apt-get upgrade

# Install Autopsy (GUI forensic browser)
sudo apt-get install autopsy

# Install Sleuth Kit (command-line forensics)
sudo apt-get install sleuthkit
```

```
# Install Guymager (disk imaging tool)
sudo apt-get install guymager

# Install Foremost (file carving)
sudo apt-get install foremost

# Install ExifTool (metadata analysis)
sudo apt-get install libimage-exiftool-perl

# Install Binwalk (embedded file detection)
sudo apt-get install binwalk

# Install Steghide (steganography detection)
sudo apt-get install steghide

# Install PhotoRec (deleted file recovery)
sudo apt-get install testdisk

# Install additional tools
sudo apt-get install scalpel
sudo apt-get install bulk-extractor
```

## Tahap 1: Akuisisi dan Preservasi Bukti Image

### 1.1 Forensic Imaging dengan dd

Tool dd adalah standar untuk membuat forensic image:[4][5]

```
# Identifikasi device yang akan di-image
sudo fdisk -l
lsblk

# Create forensic image menggunakan dd
sudo dd if=/dev/sdb of=/mnt/evidence/suspect_disk.dd bs=4M status=progress
conv=noerror,sync

# Atau menggunakan dcfldd (enhanced dd untuk forensics)
```

```
sudo dcfldd if=/dev/sdb of=/mnt/evidence/suspect_disk.dd hash=md5,sha256 bs=4M
```

**Parameter penting:**

- `if`: Input file (source device)
- `of`: Output file (destination image)
- `bs`: Block size (4M untuk kecepatan optimal)
- `conv=noerror,sync`: Continue jika ada error dan sync untuk menjaga ukuran
- `status=progress`: Menampilkan progress

## 1.2 Forensic Imaging dengan Guymager (GUI)

Guymager adalah tool GUI yang user-friendly untuk imaging:[5][6][7][8]

**Langkah-langkah:**

1. **Jalankan Guymager dengan privilege root:**

```
sudo guymager
```

2. **Di interface Guymager:**
    - Pilih device yang akan di-image (right-click)
    - Select "Acquire image"
    - Pilih format image:
        - **Linux dd raw image**: Format standar, kompatibel dengan semua tools
        - **EWF (E01)**: Expert Witness Format, mendukung compression dan metadata
        - **AFF**: Advanced Forensic Format
3. **Configure imaging options:**
    - **Image directory**: Lokasi untuk menyimpan image
    - **Image filename**: Nama file output
    - **Split image files**: Uncheck (kecuali size sangat besar)
    - **Calculate MD5**: Check untuk hash verification
    - **Calculate SHA-256**: Check untuk additional verification

- **Verify image**: Check untuk memverifikasi integritas setelah imaging

4. **Mulai proses imaging:**
    - Click "Start"
    - Guymager akan menampilkan progress dan estimated time
    - Tunggu hingga selesai dan verifikasi hash values
5. **Dokumentasi:**
    - Catat hash values (MD5 dan SHA-256)
    - Simpan log dari Guymager
    - Dokumentasikan tanggal, waktu, dan analyst name

### 1.3 Hash Verification

Verifikasi integritas image sangat penting:[9][4]

```
# Generate MD5 hash
md5sum suspect_disk.dd > suspect_disk.md5

# Generate SHA-256 hash
sha256sum suspect_disk.dd > suspect_disk.sha256

# Verify hash (setelah transfer atau sebelum analysis)
md5sum -c suspect_disk.md5
sha256sum -c suspect_disk.sha256
```

### 1.4 Write Blocking

Untuk device fisik, **SELALU gunakan write blocker** hardware atau software:[5]

```
# Software write-blocking (mount read-only)
sudo mount -o ro,loop suspect_disk.dd /mnt/evidence

# Atau mount device asli sebagai read-only
sudo mount -o ro /dev/sdb1 /mnt/evidence
```

## Tahap 2: Analisis Metadata Image

Metadata adalah informasi tersembunyi dalam file image yang dapat memberikan bukti penting.

## 2.1 Ekstraksi Metadata dengan ExifTool

ExifTool adalah tools paling powerful untuk analisis metadata:[10][11][12][13][14]

**Basic Extraction:**

```
# Ekstraksi semua metadata dari image
exiftool image.jpg

# Ekstraksi metadata dalam format lebih rapi
exiftool -common image.jpg

# Ekstraksi metadata specific tags
exiftool -Make -Model -DateTimeOriginal -GPSPosition image.jpg
```

**Advanced Analysis:**

```
# Ekstraksi semua dates dan timestamps
exiftool -AllDates -DateTimeOriginal -CreateDate -ModifyDate image.jpg

# Ekstraksi GPS coordinates
exiftool -gpslatitude -gpslongitude -gpsaltitude image.jpg
exiftool -a -G1 -gps* image.jpg

# Ekstraksi thumbnail yang embedded
exiftool -b -ThumbnailImage image.jpg > thumbnail.jpg

# Ekstraksi camera information
exiftool -Make -Model -Software -LensModel image.jpg

# Ekstraksi technical details
exiftool -ImageSize -Megapixels -ColorSpace -BitDepth image.jpg
```

**Bulk Analysis:**

```
# Analisis semua images dalam direktori
exiftool -r -ext jpg -ext png -ext gif /path/to/images/
```

```
# Export ke CSV untuk analysis
exiftool -csv -r /path/to/images/ > metadata_analysis.csv

# Recursive search dengan specific tags
exiftool -r -Artist -DateTimeOriginal -GPSPosition /evidence/photos/
```

**Metadata Manipulation Detection:**

```
# Compare file modification time dengan EXIF date
exiftool -filemodifydate -datetimeoriginal -createdate image.jpg

# Ketidakkonsistenan dapat mengindikasikan manipulasi atau editing
# Check untuk software used
exiftool -Software -CreatorTool -HistoryAction image.jpg
```

**Important Metadata Fields untuk Forensics:**

1. **Camera Information**:[11][10]
    - Make/Model: Merek dan model kamera
    - Serial Number: Nomor seri kamera (jika ada)
    - Lens Model: Lensa yang digunakan
2. **Temporal Information**:
    - DateTimeOriginal: Kapan foto diambil
    - CreateDate: Kapan file dibuat
    - ModifyDate: Kapan file terakhir dimodifikasi
    - FileModifyDate: System modification date
3. **Geolocation Data**:[10][11]
    - GPSLatitude/GPSLongitude: Koordinat lokasi
    - GPSAltitude: Ketinggian
    - GPSDateStamp/GPSTimeStamp: Timestamp GPS
4. **Technical Parameters**:
    - ISO: ISO setting

- ExposureTime: Shutter speed
- FNumber: Aperture
- FocalLength: Focal length

5. **Software/Editing Traces**:[11][10]
   - Software: Software yang digunakan untuk edit
   - Creator Tool: Tool pembuat/editor
   - History: Edit history (jika ada)

**Removing Metadata (untuk privacy atau testing):**

```
# Remove all metadata
exiftool -all= image.jpg

# Remove specific tags
exiftool -GPS*= image.jpg

# Remove metadata from multiple files
exiftool -all= *.jpg
```

## 2.2 Analyze dengan Strings

Cari text strings dalam image file untuk hidden data:[15][16]

```
# Basic strings extraction
strings image.jpg

# Strings dengan minimum length
strings -n 8 image.jpg

# Search untuk specific patterns
strings image.jpg | grep -i "password\|flag\|secret\|key"

# Strings dengan encoding
strings -e l image.jpg  # 16-bit little endian
strings -e b image.jpg  # 16-bit big endian
```

## Tahap 3: Deteksi Manipulasi Image

## 3.1 Error Level Analysis (ELA)

ELA adalah teknik untuk mendeteksi manipulasi dengan menganalisis compression artifacts:[17][18][19][20][21][22][23][24]

**Konsep ELA:**

Error Level Analysis bekerja dengan cara:

1. Menyimpan image dengan kualitas JPEG yang lebih rendah
2. Menghitung perbedaan antara original dan re-compressed image
3. Bagian yang telah diedit akan memiliki error level berbeda karena compression history yang berbeda

**Online Tools untuk ELA:**

1. **FotoForensics.com**:[18][21][22]

```
https://fotoforensics.com
```

- Upload image
- Pilih "Error Level Analysis"
- Analyze hasil: Area dengan brightness berbeda signifikan mengindikasikan manipulasi

2. **Forensically (29a.ch)**:[22][17][18]

```
https://29a.ch/photo-forensics/
```

- Upload image
- Gunakan berbagai tools:
    - Magnifier: Zoom untuk detail
    - Clone Detection: Deteksi copy-move
    - Error Level Analysis
    - Luminance Gradient
    - Principal Component Analysis

3. **ImageForensic.org**:[25][18][22]

```
http://imageforensic.org/
```

**Manual ELA dengan Python:**

```
from PIL import Image
import numpy as np

def ela_analysis(image_path, quality=90):
    # Load original image
    original = Image.open(image_path)

    # Resave with specific quality
    temp_filename = 'temp.jpg'
    original.save(temp_filename, 'JPEG', quality=quality)
    temp = Image.open(temp_filename)

    # Calculate difference
    original_array = np.array(original).astype(np.float)
    temp_array = np.array(temp).astype(np.float)

    diff = (original_array - temp_array)

    # Calculate ELA
    ela = np.abs(diff)

    # Scale to 0-255
    ela = (ela * 255 / ela.max()).astype(np.uint8)

    # Save ELA image
    ela_image = Image.fromarray(ela)
    ela_image.save('ela_output.jpg')

    return ela_image

# Usage
ela_analysis('suspicious_image.jpg', quality=90)
```

**Interpretasi ELA Results:**[20][21][23][24][22]

- **Area dengan brightness tinggi**: Kemungkinan baru ditambahkan atau diedit
- **Area dengan brightness rendah**: Original atau tidak diedit
- **Edges yang tajam**: Natural boundaries
- **Edges yang tidak konsisten**: Tanda manipulasi

**Limitations ELA:**

- Hanya efektif untuk JPEG images
- Tidak bisa mendeteksi manipulasi jika seluruh image di-save ulang dengan quality yang sama
- Hasil bisa misleading jika tidak diinterpretasi dengan benar

## 3.2 JPEG Ghost Analysis

JPEG Ghost detection mengidentifikasi area yang di-compress dengan quality factor berbeda:[26][27][28][29][30][31]

**Konsep JPEG Ghost:**

Ketika image di-composite dari multiple sources dengan different JPEG quality levels, setiap part akan memiliki compression artifacts berbeda. Ghost analysis mengekspose perbedaan ini.

**Manual Ghost Detection:**

1. Re-save image dengan berbagai quality levels (10-100)
2. Hitung difference antara original dan re-saved
3. Area yang ter-tamper akan "ghost" (muncul dengan error level berbeda) pada specific quality levels

**Online Tools:**

- Forensically.com
- GIMP with forensics plugins (untuk advanced users)

## 3.3 Clone Detection (Copy-Move Forgery)

Clone detection mengidentifikasi area dalam image yang di-copy dan paste ke location lain:[32][33][34][35][17]

**Online Clone Detection:**

**Forensically.com Clone Detector:**[17]

1. Upload image ke https://29a.ch/photo-forensics/
2. Select "Clone Detection"
3. Adjust parameters:
    - **Similarity**: Threshold untuk matching (default 80-90%)
    - **Opacity**: Transparency layer

**Manual Clone Detection dengan Python:**

```
import cv2
import numpy as np

def detect_clones(image_path):
    # Read image
    img = cv2.imread(image_path)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Initialize SIFT detector
    sift = cv2.SIFT_create()

    # Detect keypoints and descriptors
    kp, des = sift.detectAndCompute(gray, None)

    # Match descriptors with themselves
    bf = cv2.BFMatcher()
    matches = bf.knnMatch(des, des, k=2)

    # Find duplicate regions
    clones = []
    for m, n in matches:
        if m.distance < 0.7 * n.distance and m.trainIdx != m.queryIdx:
            clones.append(m)

    # Draw matches
    result = cv2.drawMatches(img, kp, img, kp, clones[:50], None,
                             flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)

    cv2.imwrite('clone_detection.jpg', result)
    return len(clones)
```

```
# Usage
num_clones = detect_clones('suspicious.jpg')
print(f"Potential clones detected: {num_clones}")
```

**Karakteristik Clone Detection:**[33][34][35]

- Deteksi duplication dalam same image
- Robust terhadap transformations (rotation, scaling, flipping)
- Common untuk hide unwanted objects atau duplicate objects

## 3.4 Splicing Detection

Splicing adalah menggabungkan parts dari different images:[36][19][37]

**Detection Methods:**

1. **Inconsistent Lighting/Shadows**:
    - Analyze light direction consistency
    - Check shadow angles dan lengths
    - Evaluate brightness distribution
2. **Inconsistent Noise Patterns**:
    - Different cameras produce different noise
    - Analyze noise level across regions
3. **Inconsistent Compression Levels**:
    - Different JPEG quality factors
    - Use JPEG Ghost analysis
4. **Color/Chromatic Aberration Inconsistencies**:
    - Edges should have consistent chromatic aberration
    - Different cameras produce different aberrations

**Forensically Tools untuk Splicing:**[17]

- **Luminance Gradient**: Analyze brightness changes

- **PCA (Principal Component Analysis)**: Reveal hidden inconsistencies

### 3.5 Camera Fingerprint Analysis (PRNU)

Photo Response Non-Uniformity (PRNU) adalah unique sensor fingerprint setiap camera:[38][39][40][41][42]

**Konsep:**

- Setiap camera sensor memiliki manufacturing imperfections unik
- PRNU adalah noise pattern yang consistent dalam semua foto dari camera yang sama
- Dapat digunakan untuk link image dengan specific camera

**Commercial Tools:**

- MOBILedit Camera Ballistics[39]
- Specialized forensic suites

**Use Cases:**

- Verify if image taken by specific camera
- Link multiple images to same source camera
- Detect if image came from suspected device

## Tahap 4: Deteksi Steganografi dalam Images

Steganography menyembunyikan data dalam images tanpa mengubah tampilan visual.

### 4.1 Visual Inspection dengan Hex Editor

```
# View hex dump
xxd image.jpg | less
hexdump -C image.jpg | less

# Check magic numbers (file signature)
xxd image.jpg | head -n 10

# JPEG should start with: FF D8 FF
# PNG should start with: 89 50 4E 47
```

### 4.2 Binwalk - Embedded File Detection

Binwalk mendeteksi files yang di-embed dalam images:[43][44][16][45][46][2]

**Basic Usage:**

```
# Scan untuk embedded files
binwalk image.jpg

# Extract embedded files automatically
binwalk -e image.jpg

# Extract with verbose mode
binwalk -Me image.jpg

# Custom extraction
binwalk --dd='.*' image.jpg
```

**Advanced Binwalk:**

```
# Scan dengan signature detection
binwalk -B image.jpg

# Entropy analysis (detect compression/encryption)
binwalk -E image.jpg

# Recursive extraction (matryoshka mode)
binwalk -Me image.jpg
```

**Interpretasi Results:**

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data
14326         0x37F6          Zip archive data
```

Jika menemukan zip, rar, atau file lain embedded, extract dengan:

```
binwalk -e image.jpg
cd _image.jpg.extracted/
```

### 4.3 Steghide - LSB Steganography

Steghide menyembunyikan data menggunakan Least Significant Bit:[45][47][43]

**Detection dan Extraction:**

```
# Extract data dari image (perlu password)
steghide extract -sf image.jpg

# Jika tidak ada password
steghide extract -sf image.jpg -p ""

# Get info tentang embedded data tanpa extract
steghide info image.jpg

# Brute force dengan wordlist (jika ada password)
for pass in $(cat wordlist.txt); do
    steghide extract -sf image.jpg -p "$pass" -xf output.txt 2>/dev/null && echo
"Password found: $pass" && break
done
```

**Creating Test Steganography:**

```
# Hide data dalam image
echo "Secret message" > secret.txt
steghide embed -cf cover.jpg -ef secret.txt -p password123
```

### 4.4 Zsteg - PNG & BMP LSB Analysis

Zsteg specialized untuk PNG dan BMP LSB steganography:[45]

```
# Install zsteg
gem install zsteg

# Analyze PNG/BMP
zsteg image.png
```

```
# Extract all LSB data
zsteg -a image.png

# Specific LSB plane
zsteg image.png --lsb
```

### 4.5 StegSolve (Java Tool)

StegSolve adalah GUI tool untuk analisis steganografi visual:

```
# Download dan run
wget http://www.caesum.com/handbook/Stegsolve.jar
java -jar Stegsolve.jar
```

**Features:**

- **Bit plane analysis**: View individual bit planes
- **Data extract**: Extract LSB data
- **Frame browser**: For animated images
- **Image combiner**: Overlay multiple images

### 4.6 Spectral Analysis untuk Hidden Messages

Messages dapat disembunyikan dalam spectrum domain:[48]

**Menggunakan Sonic Visualiser atau Audacity:**

Untuk images yang memiliki audio (rare), atau convert image data ke audio:

```
from PIL import Image
import numpy as np

# Load image
img = Image.open('image.png')
pixels = np.array(img)

# Check for patterns in specific color channels
red_channel = pixels[:,:,0]
```

```
# Analyze for hidden patterns
```

## Tahap 5: File System Forensics untuk Image Recovery

### 5.1 Sleuth Kit Analysis

Sleuth Kit adalah collection of command-line tools untuk filesystem forensics:[49][50][51][52]

**Analyze Disk Image Structure:**

```
# List partitions
mmls disk_image.dd

# File system information
fsstat -o 63 disk_image.dd

# List files and directories (including deleted)
fls -r disk_image.dd

# Display file content
icat disk_image.dd 123 > recovered_file.jpg

# Timeline analysis
fls -m / -r disk_image.dd > timeline.csv
mactime -b timeline.csv -d > timeline_human.txt
```

**Deleted File Recovery:**

```
# List deleted files
fls -d disk_image.dd

# Recover deleted file by inode
icat disk_image.dd 456 > recovered_image.jpg

# Recover all deleted JPEG files
for inode in $(fls -d disk_image.dd | grep ".jpg" | awk '{print $2}' | cut -d: -f1); do
    icat disk_image.dd $inode > recovered_$inode.jpg
done
```

## 5.2 Autopsy - Graphical Forensic Browser

Autopsy adalah GUI untuk Sleuth Kit:[53][54][55][56][57][3][58]

**Langkah-langkah Analysis:**

1. **Start Autopsy:**

```
sudo autopsy
# Default: http://localhost:9999/autopsy
```

2. **Create New Case:**
    - Case Name: Evidence_2025_001
    - Case Description: Cybercrime investigation
    - Investigators: [Your name]
3. **Add Host:**
    - Host Name: Suspect_Computer
    - Description: Windows 10 laptop
    - Timezone: Local
4. **Add Image File:**
    - Import Type: Image File
    - Image File Path: /evidence/suspect_disk.dd
    - Import Method: Symlink
    - File System Type: Auto-detect
5. **Analyze Files:**

**File Analysis Options:**

- **All Files**: View all files including deleted
- **Deleted Files**: Filter untuk deleted only
- **File Type**: Filter by extension (jpg, png, gif)
- **Keyword Search**: Search untuk filenames atau content

**Timeline Analysis:**

- View file access/modify/creation times
- Identify suspicious activity patterns
- Create timeline reports

**Hash Analysis:**

- Generate MD5 hashes untuk all files
- Compare dengan known good/bad hash databases
- Identify duplicate files

**Data Carving:**

- Recover fragmented files
- Extract files from unallocated space
- PhotoRec integration

6. **Generate Report:**
    - HTML Report
    - XML Export
    - CSV Data export

## 5.3 Foremost - File Carving

Foremost recovers files berdasarkan file signatures (headers dan footers):[59][60][61][62][63][15]

**Basic Usage:**

```
# Recover all supported file types
sudo foremost -i disk_image.dd -o /output/foremost_recovery/

# Recover specific file types
sudo foremost -t jpg,png,gif,pdf -i disk_image.dd -o /output/recovery/

# Verbose mode
sudo foremost -v -t jpg,png -i disk_image.dd -o /output/

# Work on physical device (dengan write-blocker!)
```

```
sudo foremost -t all -i /dev/sdb -o /mnt/recovery/
```

**Configuration File:**

Edit `/etc/foremost.conf` untuk customize file signatures:

```
# Tambahkan custom signatures
jpg y 200000000 \xff\xd8\xff\xe0\x00\x10 \xff\xd9
png y 200000000 \x89\x50\x4e\x47 \xff\xfc\xfd\xfe
```

**Output Analysis:**

Foremost creates output directory dengan:

- **audit.txt**: Log semua recovered files
- **Subdirectories**: One per file type (jpg/, png/, etc.)
- **Numbered files**: Recovered files dengan sequential numbers

```
# View recovery log
cat /output/foremost_recovery/audit.txt

# Count recovered files
ls -l /output/foremost_recovery/jpg/ | wc -l
```

## 5.4 PhotoRec - Advanced File Recovery

PhotoRec adalah powerful file recovery tool dari CGSecurity:[64][65][66][67][68]

**Interactive Mode:**

```
# Launch PhotoRec
sudo photorec

# Steps:
# 1. Select disk/partition
# 2. Choose partition table type (Intel/Mac/None)
# 3. Select filesystem type (ext2/ext3/ext4/Other)
# 4. Choose: Whole partition or Free space only
# 5. Select destination directory untuk recovered files
```

```
# 6. Press 'C' to start recovery
```

**Command-Line Mode:**

```
# Recover from disk image
sudo photorec /d /output/recovery/ /log disk_image.dd

# Recover specific file types
sudo photorec /d /output/ /cmd disk_image.dd
fileopt,everything,disable,fileopt,jpg,enable,fileopt,png,enable,search
```

**Best Practices PhotoRec:**[67][68][64]

1. **NEVER recover to same disk** yang sedang di-recover
2. **Prepare large storage** untuk recovered files
3. **Use file type filters** untuk speed up recovery
4. **Review audit log** untuk statistics

**PhotoRec Output:**

```
recup_dir.1/
   ├─── report.xml
   ├─── f0000001.jpg
   ├─── f0000002.png
   ├─── f0000003.jpg
   └─── ...
```

Files recovered tanpa original filenames, perlu manual review.

## 5.5 Scalpel - Alternative File Carving

Scalpel adalah enhanced version dari Foremost:[2]

```
# Install scalpel
sudo apt-get install scalpel

# Configure file types in /etc/scalpel/scalpel.conf
sudo nano /etc/scalpel/scalpel.conf
```

```
# Uncomment file types yang ingin di-recover
# jpg   y   200000000   \xff\xd8\xff\xe0\x00\x10  \xff\xd9
# png   y   20000000    \x89\x50\x4e\x47          \x49\x45\x4e\x44\xae\x42\x60\x82

# Run scalpel
sudo scalpel -b -o /output/scalpel/ disk_image.dd
```

## Tahap 6: Advanced Image Analysis Techniques

### 6.1 Batch Processing dengan Scripts

**Python Script untuk Bulk Metadata Extraction:**

```
#!/usr/bin/env python3
import os
import subprocess
import csv

def extract_metadata_bulk(directory):
    """Extract metadata from all images in directory"""
    results = []

    for root, dirs, files in os.walk(directory):
        for file in files:
            if file.lower().endswith(('.jpg', '.jpeg', '.png', '.gif')):
                filepath = os.path.join(root, file)

                # Run exiftool
                cmd = f"exiftool -DateTimeOriginal -Make -Model -GPSPosition {filepath}"
                output = subprocess.check_output(cmd, shell=True).decode()

                results.append({
                    'filename': file,
                    'path': filepath,
                    'metadata': output
                })

    # Save to CSV
    with open('metadata_report.csv', 'w', newline='') as f:
```

```
        writer = csv.DictWriter(f, fieldnames=['filename', 'path', 'metadata'])
        writer.writeheader()
        writer.writerows(results)

    return results

# Usage
extract_metadata_bulk('/evidence/images/')
```

**Bash Script untuk Automated Analysis:**

```
#!/bin/bash
# Image Forensics Automation Script

IMAGE_DIR="/evidence/images"
OUTPUT_DIR="/evidence/output"

mkdir -p $OUTPUT_DIR/{metadata,ela,clone_detection,recovered}

echo "[*] Starting Image Forensics Analysis..."

# 1. Metadata Extraction
echo "[*] Extracting metadata..."
exiftool -r -csv $IMAGE_DIR > $OUTPUT_DIR/metadata/all_metadata.csv

# 2. Hash all images
echo "[*] Hashing all images..."
find $IMAGE_DIR -type f \( -iname "*.jpg" -o -iname "*.png" \) -exec sha256sum {} \; >
$OUTPUT_DIR/image_hashes.txt

# 3. Binwalk scan
echo "[*] Scanning for embedded files..."
for img in $(find $IMAGE_DIR -type f); do
    binwalk $img >> $OUTPUT_DIR/binwalk_results.txt
done

# 4. Strings extraction
echo "[*] Extracting strings..."
for img in $(find $IMAGE_DIR -type f); do
    echo "=== $img ===" >> $OUTPUT_DIR/strings_output.txt
```

```
    strings -n 8 $img >> $OUTPUT_DIR/strings_output.txt
done

echo "[*] Analysis complete. Results in $OUTPUT_DIR"
```

## 6.2 Image Comparison dan Similarity Detection

**ImageMagick Compare:**

```
# Install ImageMagick
sudo apt-get install imagemagick

# Compare two images
compare image1.jpg image2.jpg diff.png

# Calculate difference percentage
compare -metric RMSE image1.jpg image2.jpg null: 2>&1

# Highlight differences
compare image1.jpg image2.jpg -compose src diff_highlight.png
```

**Python Image Similarity:**

```
from PIL import Image
import imagehash

def compare_images(img1_path, img2_path):
    """Compare two images using perceptual hashing"""
    hash1 = imagehash.average_hash(Image.open(img1_path))
    hash2 = imagehash.average_hash(Image.open(img2_path))

    difference = hash1 - hash2

    if difference == 0:
        return "Identical"
    elif difference < 5:
        return "Very Similar"
    elif difference < 10:
        return "Similar"
```

```
    else:
        return "Different"

# Usage
result = compare_images('photo1.jpg', 'photo2.jpg')
print(result)
```

### 6.3 Reverse Image Search

```
# Menggunakan TinEye API atau Google Images
# Untuk find source/original dari manipulated image

# Manual: Upload ke
# - https://tineye.com
# - https://images.google.com
# - https://yandex.com/images
```

## Tahap 7: Dokumentasi dan Pelaporan

### Chain of Custody Form

Dokumentasi yang proper sangat critical untuk admissibility di pengadilan:

**Information yang harus didokumentasikan:**

1. **Evidence Identification**
    - Case Number
    - Evidence ID/Tag
    - Description of evidence
    - Source location
    - Date/Time of collection
2. **Acquisition Details**
    - Acquisition method (dd, Guymager, etc.)
    - Tools dan versions used

- Hash values (MD5, SHA-256)
- Disk size dan partitions
- Any issues encountered

3. **Chain of Custody**
   - Analyst name
   - Date/Time of transfer
   - Purpose of transfer
   - Storage location
   - Access log
4. **Analysis Actions**
   - Tools used dengan versions
   - Commands executed
   - Findings dan observations
   - Timestamps for all actions

## Forensic Report Structure

```
# Digital Image Forensics Report

## Case Information
- Case Number: 2025-CYBER-001
- Investigator: [Name]
- Date of Analysis: 2025-10-20
- Evidence Item: IMG_001.jpg

## Executive Summary
Brief summary of findings and conclusions.

## Evidence Description
- File Name: IMG_001.jpg
- File Size: 2.4 MB
- MD5 Hash: d41d8cd98f00b204e9800998ecf8427e
- SHA-256 Hash: e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

```
## Methodology
Tools and techniques used for analysis:
1. Metadata extraction dengan ExifTool v12.40
2. ELA analysis via FotoForensics
3. Clone detection dengan Forensically
4. Steganography detection dengan Binwalk dan Steghide

## Findings

### Metadata Analysis
- Camera Make: Canon
- Camera Model: EOS 5D Mark IV
- Date Taken: 2025-10-15 14:32:10
- GPS Location: [coordinates]
- Software: Adobe Photoshop CS6

### Manipulation Detection
- ELA Results: High error level detected in [specific region]
- Interpretation: Evidence of image editing in [area]

### Steganography Analysis
- Binwalk: No embedded files detected
- Steghide: No hidden data found

## Timeline
- 2025-10-15 14:32:10: Photo captured
- 2025-10-16 09:15:22: File modified (per filesystem)
- 2025-10-16 09:20:45: Last access

## Conclusions
Based on forensic analysis, the image shows evidence of manipulation in [specific
areas]. The metadata indicates use of Adobe Photoshop after original capture. GPS
coordinates place the photo at [location].

## Limitations
- Analysis conducted on copy, not original device
- No access to original camera for PRNU comparison
- Limited by JPEG compression artifacts

## Appendices
```

```
- A: Complete metadata dump
- B: ELA analysis images
- C: Hash verification logs
- D: Tool screenshots

## Certification
I certify that this analysis was conducted following accepted forensic practices and
that the findings accurately represent the evidence examined.

[Signature]
[Date]
```

### Generating Report dengan Python

```
#!/usr/bin/env python3
import os
from datetime import datetime
import subprocess

def generate_forensic_report(image_path, output_path):
    """Generate comprehensive forensic report"""

    report = []
    report.append("=" * 80)
    report.append("DIGITAL IMAGE FORENSICS REPORT")
    report.append("=" * 80)
    report.append(f"\nGenerated: {datetime.now()}")
    report.append(f"Image File: {image_path}\n")

    # File info
    report.append("\n[FILE INFORMATION]")
    stat = os.stat(image_path)
    report.append(f"File Size: {stat.st_size} bytes")
    report.append(f"Modified: {datetime.fromtimestamp(stat.st_mtime)}")

    # Hash values
    report.append("\n[HASH VALUES]")
    md5 = subprocess.check_output(f"md5sum {image_path}",
shell=True).decode().split()[^0]
```

```
    sha256 = subprocess.check_output(f"sha256sum {image_path}",
shell=True).decode().split()[^0]
    report.append(f"MD5: {md5}")
    report.append(f"SHA-256: {sha256}")

    # Metadata
    report.append("\n[METADATA ANALYSIS]")
    metadata = subprocess.check_output(f"exiftool {image_path}", shell=True).decode()
    report.append(metadata)

    # Binwalk
    report.append("\n[EMBEDDED FILE SCAN]")
    try:
        binwalk_output = subprocess.check_output(f"binwalk {image_path}", shell=True,
stderr=subprocess.DEVNULL).decode()
        report.append(binwalk_output if binwalk_output else "No embedded files
detected")
    except:
        report.append("No embedded files detected")

    # Strings
    report.append("\n[STRING ANALYSIS]")
    strings = subprocess.check_output(f"strings -n 10 {image_path} | head -n 20",
shell=True).decode()
    report.append(strings)

    # Save report
    report_text = "\n".join(report)
    with open(output_path, 'w') as f:
        f.write(report_text)

    print(f"[+] Report saved to: {output_path}")
    return report_text

# Usage
generate_forensic_report('/evidence/suspect_image.jpg', '/evidence/forensic_report.txt')
```

## Best Practices Image Forensics

## Do's:

1. **Always work on copies**[1][5]
    - Never analyze original evidence directly
    - Create forensic images dengan verified hashes
    - Use write-blockers untuk physical media
2. **Document everything**[4]
    - Maintain detailed chain of custody
    - Log all commands dan tools used
    - Screenshot important findings
    - Record timestamps untuk semua actions
3. **Verify integrity**[9][4][5]
    - Calculate hashes before dan after transfer
    - Verify hashes throughout analysis
    - Document any integrity issues
4. **Use multiple tools**[69][18]
    - Cross-verify findings dengan different tools
    - Different tools have different strengths
    - Reduce false positives/negatives
5. **Boot forensic mode**[1]
    - Use Kali Linux forensic mode
    - Prevent auto-mounting
    - Disable swap
6. **Understand tool limitations**[69]
    - Know what each tool can dan cannot do
    - Interpret results correctly
    - Don't over-rely on automated tools

7. **Keep updated**
    - Update forensic tools regularly
    - Stay current dengan new techniques
    - Follow forensics community
8. **Preserve metadata**
    - Don't strip metadata during analysis
    - Backup original metadata
    - Document metadata changes

## Don'ts:

1. **Don't modify originals**[1]
    - Never edit original evidence
    - No "quick checks" on originals
    - Always use copies
2. **Don't assume**
    - Don't assume image is authentic without analysis
    - Don't assume metadata is accurate (can be spoofed)[10]
    - Don't assume tools are 100% accurate
3. **Don't ignore context**
    - Consider full investigation context
    - Metadata alone doesn't prove authenticity
    - Look for corroborating evidence
4. **Don't skip verification**
    - Always verify hashes
    - Verify tool outputs
    - Cross-check findings
5. **Don't exceed expertise**

- Know your limitations
- Consult specialists when needed
- Don't testify beyond qualifications

6. **Don't forget legal requirements**
   - Follow jurisdiction-specific laws
   - Maintain admissibility standards
   - Respect privacy regulations
7. **Don't contaminate evidence**
   - Use clean tools dan storage
   - Prevent cross-contamination
   - Isolate analyzed evidence
8. **Don't make unsupported conclusions**
   - Base conclusions on evidence
   - Acknowledge limitations
   - Present alternative explanations

## Kesimpulan

Image forensics adalah bidang yang kompleks dan terus berkembang. Dengan tools yang tersedia di Kali Linux dan pemahaman teknik yang proper, investigator dapat:

- **Memverifikasi keaslian** gambar digital
- **Mendeteksi manipulasi** menggunakan ELA, JPEG Ghost, clone detection
- **Mengekstrak metadata** untuk informasi temporal, geolocation, dan device
- **Mendeteksi steganography** menggunakan binwalk, steghide, dan analisis manual
- **Me-recover deleted images** dengan foremost, photorec, dan autopsy
- **Menganalisis filesystem** untuk reconstruct aktivitas
- **Generate comprehensive reports** untuk legal proceedings

**Key Takeaways:**

1. **Preservasi adalah prioritas** - Always work on forensically sound copies[5][1]
2. **Dokumentasi adalah kunci** - Maintain meticulous records untuk admissibility[4]
3. **Multiple tools, multiple techniques** - Cross-verify findings[18][69]
4. **Context matters** - Interpret results dalam full investigation context
5. **Continuous learning** - Field terus evolve dengan new techniques dan challenges[19][36]

Dengan mengikuti tutorial step-by-step ini, Anda dapat melakukan image forensics untuk investigasi cybercrime menggunakan Kali Linux secara professional dan forensically sound.[54][60][56][3][12][18][2][17][10]

⁂


1. https://www.kali.org/docs/general-use/kali-linux-forensics-mode/
2. https://www.geeksforgeeks.org/linux-unix/kali-linux-forensics-tools/
3. https://www.infosecinstitute.com/resources/digital-forensics/kali-linux-top-5-tools-for-digital-forensics/
4. https://www.scribd.com/document/332492199/Tutorial-6-Kali-Linux-Sleuthkit
5. https://www.cybrary.it/blog/creating-a-forensic-disk-image-using-the-linux-guymager-utility
6. https://guymager.sourceforge.io
7. https://awjunaid.com/kali-linux/guymager-a-forensic-imaging-tool-for-creating-disk-images-and-performing-hash-verification/
8. https://www.kali.org/tools/guymager/
9. https://hackers-arise.com/digital-forensics-part-1-capturing-a-forensically-sound-image/
10. https://techarry.com/using-exiftool-in-kali-linux-for-metadata-extraction/
11. https://www.osintteam.com/using-exiftool-to-extract-metadata-from-image-files/
12. https://www.hackingarticles.in/exiftool-a-meta-data-extractor/
13. https://www.kali.org/tools/libimage-exiftool-perl/
14. https://www.hackingarticles.in/forensic-investigation-examine-corrupt-file-metadata/
15. https://samsclass.info/121/proj/p6-fore.htm

16. https://github.com/cyb0rgdoll/image-steg
17. https://29a.ch/photo-forensics/
18. https://www.telkomnika.uad.ac.id/index.php/TELKOMNIKA/article/download/15295/9288
19. https://pmc.ncbi.nlm.nih.gov/articles/PMC11323046/
20. https://en.wikipedia.org/wiki/Error_level_analysis
21. https://bumidatar.id/fotoforensics
22. https://ejurnal.seminar-id.com/index.php/tin/article/download/859/581/
23. https://www.academia.edu/45677695/Forensik_Foto_Digital_dengan_Teknik_Error_Level_Analysis_ELA
24. https://catatanforensikadigital.wordpress.com/2018/05/12/ela-error-level-analysis/
25. https://www.imageforensic.org
26. https://sourceforge.net/p/gimp-forensics/wiki/JPEG Ghost/
27. https://pmc.ncbi.nlm.nih.gov/articles/PMC6839440/
28. https://farid.berkeley.edu/downloads/publications/tifs09.pdf
29. https://www5.cs.fau.de/forschung/groups/computer-vision/image-forensics/detection-of-jpeg-ghosts/index.html
30. https://www.diva-portal.org/smash/get/diva2:1503622/FULLTEXT02
31. https://dl.acm.org/doi/abs/10.1007/s11042-022-13699-x
32. https://sites.google.com/site/elsamuko/forensics/clone-detection
33. http://lci.micc.unifi.it/labd/2015/01/copy-move-forgery-detection-and-localization/
34. https://core.ac.uk/download/pdf/199242518.pdf
35. https://www.sciencedirect.com/science/article/pii/S1319157822004323
36. https://imagetwin.ai/image-manipulation-detection
37. https://www.klippa.com/en/blog/information/image-tampering-detection/
38. http://ws2.binghamton.edu/fridrich/Research/full_paper_02.pdf
39. https://www.mobiledit.com/camera-ballistics-details

40. https://ieeexplore.ieee.org/document/4712000

41. https://dl.acm.org/doi/10.1145/3576915.3616600

42. https://arxiv.org/abs/2111.02144

43. https://www.linkedin.com/pulse/detect-steganography-digital-media-tools-techniques-you-pandey-w4hkf

44. https://ijcsmc.com/docs/papers/June2021/V10I6202112.pdf

45. https://infosecwriteups.com/some-common-steganography-tools-for-ctfs-92e3de93f141

46. https://www.wattlecorp.com/top-3-steganography-tools/

47. https://0xrick.github.io/lists/stego/

48. https://www.rcs.cic.ipn.mx/2023_152_10/Steganography in Frequency Domain_ Hiding Text through Audio Spectrogram.pdf

49. https://www.kali.org/tools/sleuthkit/

50. https://www.sleuthkit.org/sleuthkit/

51. https://www.youtube.com/watch?v=CKtTQyc_5NE

52. https://www.randylee.com/cybersecurity/kali-linux-essentials/exploring-the-extensive-toolset-of-kali-linux/forensics-tools-autopsy-and-sleuth-kit

53. https://www.linkedin.com/pulse/digital-forensics-practical-tutorial-using-autopsy-kali-handaya-mftxc

54. https://www.kali.org/tools/autopsy/

55. https://www.autopsy.com

56. https://www.geeksforgeeks.org/linux-unix/autopsy-cyber-forensic-browser-in-kali-linux/

57. https://www.autopsy.com/download/

58. https://www.sleuthkit.org/autopsy/

59. https://www.kali.org/tools/foremost/

60. https://www.hackingarticles.in/forensic-data-carving-using-foremost/

61. https://linuxconfig.org/how-to-recover-deleted-files-with-foremost-on-linux

62. http://racfor.zesoi.fer.hr/doku.php?id=racfor_wiki%3Adatoteke_i_datotecni_sustavi%3Afile_carving_with_the_foremost_tool

63. https://infosecwriteups.com/how-to-recover-deleted-files-using-foremost-in-linux-7b070ffcb307

64. https://www.cgsecurity.org/wiki/photoRec

65. https://www.cgsecurity.org/testdisk_doc/photorec.html

66. https://www.youtube.com/watch?v=c0j9jE0X4nY

67. https://www.digitalocean.com/community/tutorials/photorec-recover-deleted-files-in-linux-ubuntu

68. https://recoverit.wondershare.com/photo-recovery/how-to-use-photorec.html

69. https://commons.erau.edu/jdfsl/vol17/iss2/4/

70. https://systemweakness.com/volatility-memory-image-forensics-74ecfea17c2f

71. https://www.sciencedirect.com/science/article/abs/pii/S0379073820301730

72. https://www.bluevoyant.com/knowledge-center/get-started-with-these-9-open-source-tools

73. https://dl.acm.org/doi/10.1145/3731243

74. https://www.stationx.net/kali-linux-tutorial/

75. https://arxiv.org/abs/2203.15880

76. https://github.com/mesquidar/ForensicsTools

77. https://www.youtube.com/watch?v=8L1L-2vyfrs

78. https://www.salvationdata.com/knowledge/forensic-imaging/

79. https://www.youtube.com/watch?v=9AyiRITI9HI

80. https://www.youtube.com/watch?v=OGlRKz2PECg

81. https://www.youtube.com/watch?v=mqHx7HutQLo

82. https://www.uphop.ai/app/c/9a3b2b53-6270-47d9-89ec-30f3c6828526

83. https://www.youtube.com/watch?v=TyphqF3m2Uw

84. https://www.youtube.com/watch?v=Z0_mlmY9T4E

85. https://www.youtube.com/watch?v=8DD1FCc-yew

86. https://exiftool.org

87. https://e-modul.instiki.ac.id/matkul_sk/e_module_cyber_forensics/files/basic-html/page38.html

88. https://github.com/shanedevane/python-image-clone-detection

89. https://ieeexplore.ieee.org/document/5582952/

90. https://media.neliti.com/media/publications/427161-an-overview-of-segmentation-based-image-b77ba0f6.pdf

91. https://www.semanticscholar.org/paper/fc9f4138baffca1868672b6b740be99fb137145f

92. https://github.com/volatilityfoundation/volatility

93. https://volatilityfoundation.org

94. https://gitlab.com/kalilinux/packages/sleuthkit

95. https://www.varonis.com/blog/how-to-use-volatility

96. https://discussion.fedoraproject.org/t/recover-deleted-media-with-testdisk-photorec-and-scalpel/111313

97. https://www.hackingarticles.in/memory-forensics-using-volatility-framework/

98. https://github.com/volatilityfoundation/volatility/wiki/Memory-Samples

99. https://cyberhub.sa/posts/5835

100. http://www.sleuthkit.org/autopsy/help/file_mode.html