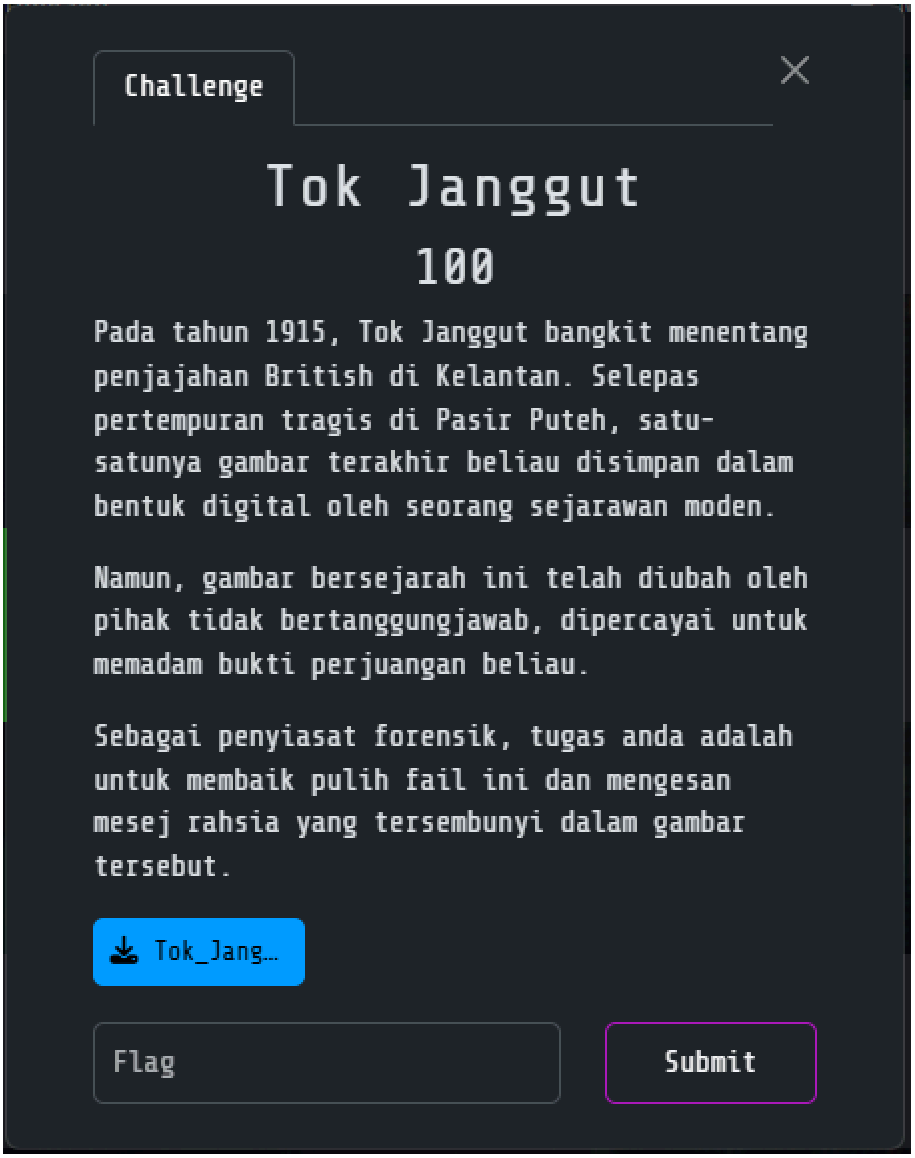
Tok Janggut

▨ Types	forensic
▨ CTF	3108

📝 **Write-Up: Forensic Challenge — *Tok Janggut (100 pts)***



Tok Janggut

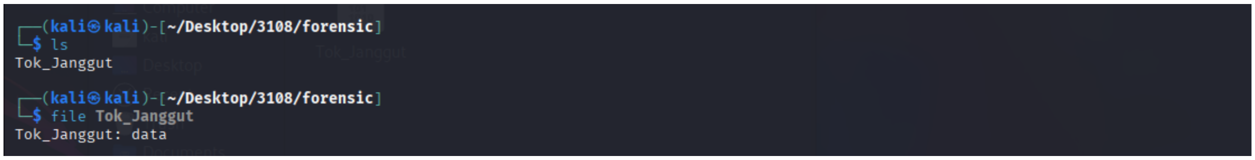
100

Pada tahun 1915, Tok Janggut bangkit menentang penjajahan British di Kelantan. Selepas pertempuran tragis di Pasir Puteh, satu-satunya gambar terakhir beliau disimpan dalam bentuk digital oleh seorang sejarawan moden.

Namun, gambar bersejarah ini telah diubah oleh pihak tidak bertanggungjawab, dipercayai untuk memadam bukti perjuangan beliau.

Sebagai penyiasat forensik, tugas anda adalah untuk membaik pulih fail ini dan mengesan mesej rahsia yang tersembunyi dalam gambar tersebut.

We were given a suspicious file named `Tok_Janggut`. Running the `file` command showed:



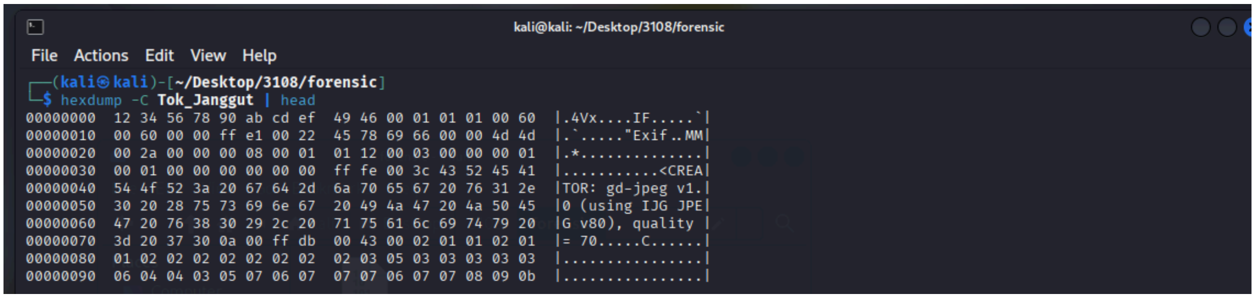
```
┌──(kali㉿kali)-[~/Desktop/3108/forensic]
└─$ ls
Tok_Janggut

┌──(kali㉿kali)-[~/Desktop/3108/forensic]
└─$ file Tok_Janggut
Tok_Janggut: data
```

This means the file has no recognized header — its magic bytes are corrupted.

Step 1 — Inspecting the File Header

I opened the file in

```
hexdump -C Tok_Janggut | head
```



kali@kali: ~/Desktop/3108/forensic
File Actions Edit View Help

```
┌──(kali㉿kali)-[~/Desktop/3108/forensic]
└─$ hexdump -C Tok_Janggut | head
00000000  12 34 56 78 90 ab cd ef  49 46 00 01 01 01 00 60  |.4Vx....IF.....`|
00000010  00 60 00 00 ff e1 00 22  45 78 69 66 00 00 4d 4d  |.`....."Exif..MM|
00000020  00 2a 00 00 00 08 00 01  01 12 00 03 00 00 00 01  |.*..............|
00000030  00 01 00 00 00 00 00 00  ff fe 00 3c 43 52 45 41  |...........<CREA|
00000040  54 4f 52 3a 20 67 64 2d  6a 70 65 67 20 76 31 2e  |TOR: gd-jpeg v1.|
00000050  30 20 28 75 73 69 6e 67  20 49 4a 47 20 4a 50 45  |0 (using IJG JPE|
00000060  47 20 76 38 30 29 2c 20  71 75 61 6c 69 74 79 20  |G v80), quality |
00000070  3d 20 37 30 0a 00 ff db  00 43 00 02 01 01 02 01  |= 70.....C......|
00000080  01 02 02 02 02 02 02 02  02 03 05 03 03 03 03 03  |................|
00000090  06 04 04 03 05 07 06 07  07 07 06 07 07 08 09 0b  |................|
```

The first 8 bytes looked like this:

```
12 34 56 78 90 AB CD EF
```

But a normal JPEG file should start with:

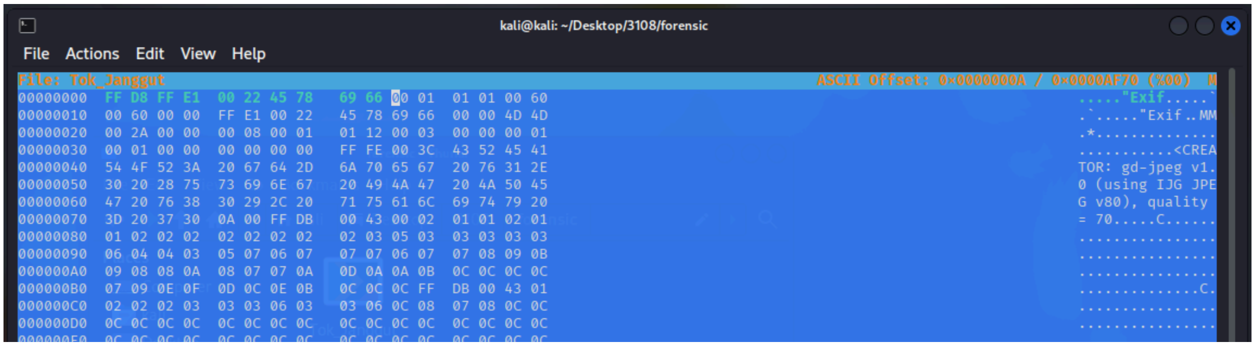
```
FF D8 FF E0  (or FF D8 FF E1 depending on marker)
```

👉 This confirmed the header was tampered with.

Step 2 — Restoring the JPEG Header

Manually, I replaced the first 8 bytes with a valid JPEG SOI (Start of Image) header:

```
FF D8 FF E0 00 10 4A 46
```



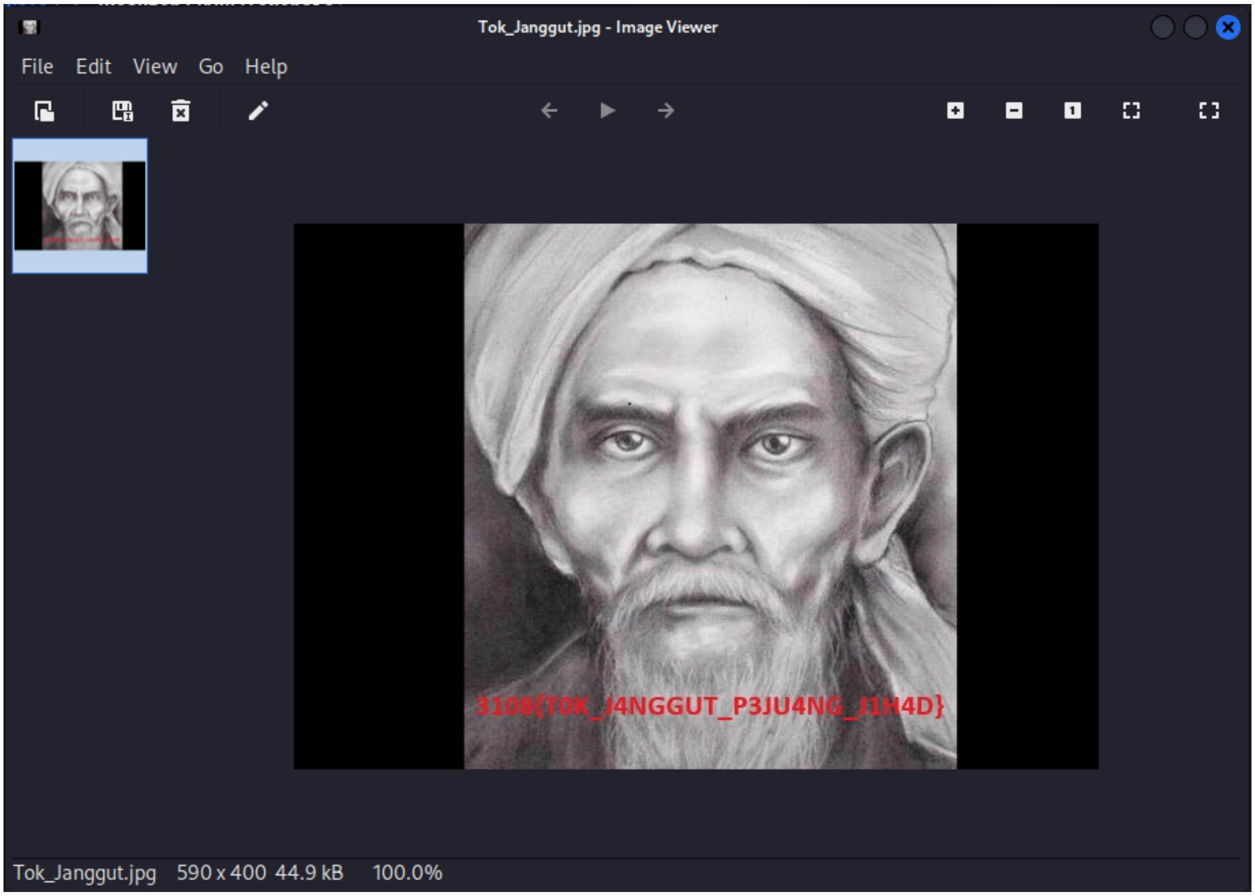
kali@kali: ~/Desktop/3108/forensic

```
File: Tok_Janggut                                        ASCII Offset: 0x0000000A / 0x0000AF70 (%00)  M
00000000  FF D8 FF E1  00 22 45 78  69 66 00 01  01 01 00 60   ....."Exif.....`
00000010  00 60 00 00  FF E1 00 22  45 78 69 66  00 00 4D 4D   .`....."Exif..MM
00000020  00 2A 00 00  00 08 00 01  01 12 00 03  00 00 00 01   .*..............
00000030  00 01 00 00  00 00 00 00  FF FE 00 3C  43 52 45 41   ............<CREA
00000040  54 4F 52 3A  20 67 64 2D  6A 70 65 67  20 76 31 2E   TOR: gd-jpeg v1.
00000050  30 20 28 75  73 69 6E 67  20 49 4A 47  20 4A 50 45   0 (using IJG JPE
00000060  47 20 76 38  30 29 2C 20  71 75 61 6C  69 74 79 20   G v80), quality 
00000070  3D 20 37 30  0A 00 FF DB  00 43 00 02  01 01 02 01   = 70.....C......
00000080  01 02 02 02  02 02 02 02  02 03 05 03  03 03 03 03   ................
00000090  06 04 04 03  05 07 06 07  07 07 06 07  07 08 09 0B   ................
000000A0  09 08 08 0A  08 07 07 0A  0D 0A 0A 0B  0C 0C 0C 0C   ................
000000B0  07 09 0E 0F  0D 0C 0E 0B  0C 0C 0C FF  DB 00 43 01   ..............C.
000000C0  02 02 02 03  03 03 06 03  03 06 0C 08  07 08 0C 0C   ................
000000D0  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C   ................
000000E0  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C
```

- FF D8 → JPEG start marker (SOI).
- FF E0 → APP0 marker.
- 00 10 → block length.
- 4A 46 → “JF” (part of “JFIF”).

After saving the modified file as Tok_Janggut_fixed.jpg, I tried to open it.

then just open the image and we get the flag


Tok_Janggut.jpg - Image Viewer
File
Edit
View
Go
Help
3108{T0K_J4NGGUT_P3JU4NG_J1H4D}
Tok_Janggut.jpg 590 x 400 44.9 kB 100.0%

✅ **Conclusion**:

This challenge tested knowledge of **file headers** and **basic file repair in forensics**. By manually restoring the JPEG magic bytes, the image was recoverable, and the hidden flag could be extracted.