Implementation of FFMPEG Video Compression to Improve Performance and Storage Efficiency in Digital Knowledge Repository System

I Made Riyan Adi Nugroho[1], Agus Adi Putrawan[2], Sri Andriati Asri[3], and Made Pradnyana Ambara[4]

[1,2,3,4] Information Technology Department, Politeknik Negeri Bali, Bali, Indonesia
maderiyan@pnb.ac.id

Abstract. The focus of this research is to implement FFmpeg video compression to improve performance and storage efficiency in the Digital Knowledge Repository System. Various applications can be used to compress video files, with FFmpeg being one of the most popular. In this study, FFmpeg was implemented as the video compression software using the PHP-FFMpeg library. For video compression, the libvpx-vp9 encoder was used for video and libopus for audio, with a video bitrate of 0.33 Mbit/s and an audio bitrate of 96 Kbit/s. The implementation of the video compression feature in this web-based Digital Knowledge Repository Application has been functionally successful, as confirmed by the results of functional testing. The effectiveness of the compression has also been demonstrated, achieving a reduction of 47.42% in file size during trials.

Keywords: FFMPEG, Knowledge Repository, Video Compression

1 Introduction

The Digital Knowledge Repository System is a service designed to manage knowledge in various formats, including text, audio, images, and video (Nugroho, 2021). Large file sizes require significant storage space, causing the system's storage to fill up quickly (Smiti et al., 2018). When storage exceeds its capacity or becomes full, the system's performance may degrade, potentially leading to system failure (Mao et al., 2018).

One way to address this issue is by reducing the file size without altering the video's duration. Video compression is an effective method for reducing file size by adjusting the video quality (Triantoro et al., 2018). Compression is achieved by lowering both the image resolution and audio quality within the video file. Reducing both image resolution and audio quality results in a smaller file size compared to reducing only the image resolution.

There are various applications available for compressing video files, one of the most popular being FFmpeg (El-Arsh et al., 2021). FFmpeg is chosen for video file

A. A. N. G. Sapteka et al. (eds.), *Proceedings of the International Conference on Sustainable Green Tourism Applied Science - Engineering Applied Science 2024 (ICoSTAS-EAS 2024)*, Advances in Engineering Research 249,
https://doi.org/10.2991/978-94-6463-587-4_43

compression not only because it is open-source but also because it is relatively easy to integrate with other applications.

In this research, FFmpeg was integrated into the Digital Knowledge Repository System as a video compression solution. The goal of this integration was clear: to compress videos before they were stored on the server, reducing the space they occupied. The process involved taking digital knowledge in video format and applying FFmpeg's compression capabilities within the system.

To test the effectiveness of this approach, file sizes were carefully compared before and after compression. This gave insight into how much storage space could be saved. In addition, the system's performance in handling and playing the compressed video files was measured, ensuring that the repository could run smoothly while maintaining efficient storage usage.

2 Methodology

The purpose of this research is to integrate FFmpeg software with the Digital Knowledge Repository System (DKRS) to reduce the size of video files managed by the system. Additionally, the study aims to evaluate the effectiveness of the compression process in improving system performance and optimizing server space usage. To achieve these objectives, the research was conducted in four stages: system requirements analysis, system design, implementation, and testing (Zeng et al., 2017).

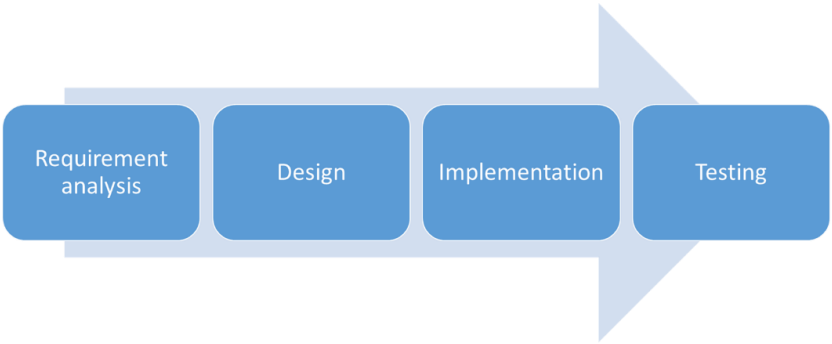

Figure 1. Research methods

2.1 System Requirements Analysis

The Digital Knowledge Repository System is a web-based application designed to manage digital knowledge, including video files. Within this system, users can upload their own digital knowledge content. Additionally, users can view or access digital knowledge uploaded by others.

The large file sizes require significant storage space and negatively impact the application's performance. To address this issue, it's necessary to reduce the video file size without altering its duration. One popular solution is video compression, which reduces file size by lowering the image resolution and audio quality of the video.

To implement the video file compression feature in DKRs, a software capable of performing video compression and integrating with DKRs is required. FFmpeg is an audio and video processing software with built-in video compression features. It is open source and offers various mechanisms for integration with other applications (FFMPEG, 2010). For this purpose, the PHP-FFMpeg library is used to integrate FFmpeg with DKRs. This library was selected because DKRs is a web-based application developed using the PHP programming language (Safin et al., 2020).

2.2 System Design

The video compression process will be implemented within the digital knowledge data upload feature of the Digital Knowledge Repository System. When a user uploads digital knowledge data, the file will first be compressed before being stored on the server. The process begins when the user navigates to the digital knowledge data upload page. On this page, the user is prompted to select the video file to upload. After the user clicks the upload button, the system will detect the file type. If the file type is compatible (e.g., mkv, avi, mp4), the system will proceed to upload the file and then perform the compression process. The compression mechanism is illustrated in Figure 2.

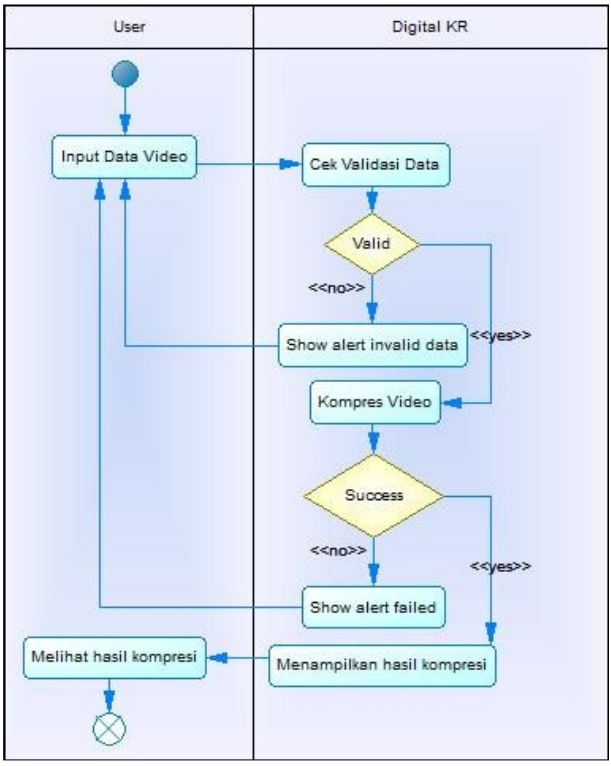

Figure 2. Video compression flow

2.3 Design Implementation

Integrating DKRs with FFmpeg is achieved by creating an integration module using the PHP-FFMpeg library. This module was developed in PHP and leverages the functions provided by FFmpeg (Kristiadi & Marwiyati, 2021; Gupta et al., 2021). Before running the integration module, FFmpeg software must be installed on the server. Figure 3 demonstrates that FFmpeg has been successfully installed on the server.

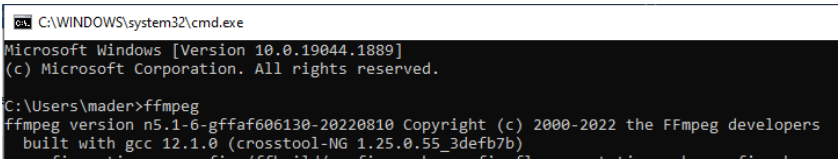

Figure 3. FFMPEG installed on the server

The video compression process is achieved by adjusting the bitrate of both the video and audio (Zeng et al., 2017). Figure 4 illustrates the function call used to modify the video and audio bitrates.

```
ffmpeg -i data1.mp4 -c:v libvpx-vp9 -b:v 0.33M -c:a libopus -b:a 96k target1-2.webm
ffmpeg -i data2.mp4 -c:v libvpx-vp9 -b:v 0.33M -c:a libopus -b:a 96k target2-2.webm
ffmpeg -i data3.mp4 -c:v libvpx-vp9 -b:v 0.33M -c:a libopus -b:a 96k target3-2.webm
ffmpeg -i data4.mp4 -c:v libvpx-vp9 -b:v 0.33M -c:a libopus -b:a 96k target4-2.webm
ffmpeg -i data5.mp4 -c:v libvpx-vp9 -b:v 0.33M -c:a libopus -b:a 96k target5-2.webm
```

Figure 4. Video compression process log

2.4 Testing

Testing is conducted to evaluate both the system's functionality and the effectiveness of the compression process. Functionality testing ensures that the compression process works correctly and that the system can display the compressed videos. The effectiveness of the compression is assessed by calculating the compression ratio, the percentage of storage space saved, and the change in access time. These metrics help determine the overall efficiency and impact of the compression process (Pavlic & Burkeljca, 2019).

3 Result and Discussion

In this research, two aspects of testing were conducted: functional testing of the compression system and testing the effectiveness of the compression process.

3.1 Functional Testing Results

The testing process was conducted on two features: the video file compression feature and the feature for displaying the compressed file. All four test scenarios produced the expected results. The details of these scenarios can be seen in Table 1.

Table 1. Functional testing results

No	Code	Testing scenario	Test case	Expected results	Testing results
1	SI001	Video file compression feature			
		No video file	File: (empty)	The system displays a message indicating that the file data cannot be empty.	as expected
		File not in video format	File: (.doc)	The system displays a message indicating that the file type must be in a video format.	as expected
		Input video format file	File: (video)	The system successfully compresses the file and displays a message confirming that the file has been successfully compressed.	as expected
2	SI002	Feature for displaying the compressed file.			
		The user selects the data to be displayed.	Select data	The system will display the selected data and then show the video in a video player.	as expected

3.2 Compression Effectiveness Test Results

Testing was conducted on five .mp4 video files, which were downloaded from YouTube. The test data included videos of various file sizes to observe the trends in compression ratio and assess the success of the compression process. The results of the compression are presented in Table 2.

The compression ratio is defined as the ratio of the original file size to the compressed file size. This value indicates the effectiveness of the compression process. A higher compression ratio means greater reduction in file size, but it is inversely proportional to the quality of the resulting video. The calculation of the video file compression ratio is given by Equation (1) (Triantoro et al., 2018):

$$Compression\ ratio\ = \frac{original\ file\ size}{compressed\ file\ size} \tag{1}$$

The percentage of savings is calculated based on the difference in file size before and after the compression process. This percentage indicates the amount of storage space saved. The calculation of the percentage savings is given by Equation (**2**) (Triantoro et al., 2018):

$$PP = 100\% - \left(\frac{O}{I} x\ 100\%\right) \quad (2)$$

Description:
PP : Percentage of Savings
O : Compressed file size (MB)
I : Original file size (MB)

Table 2. Compression effectiveness test results

No	Size (before)	Size (after)	Ratio	% Savings	Time
1	18.6 MB	9 MB	2.07	52	1:04
2	79.7 MB	6.61 MB	12.06	92	12:47
3	13.6 MB	7.46 MB	1.82	45	1:52
4	10.1 MB	5.86 MB	1.72	42	2:59
5	4.37 MB	4.08 MB	1.07	7	2:46
Average			3.75	47.42	

The results show that the compression ratio for the five data samples ranges from 1.07 to 12.06, with an average of 3.75. The percentage of storage space saved varies from 7% to 92%, with an average of 47.42%. Based on this data, it can be concluded that the compression ratio is directly proportional to the percentage of space saved.

The factors influencing the compression process include the encoder used, video and audio bitrate, and video resolution. In this trial, the encoder used for video is libvpx-vp9, and for audio, it is libopus. The bitrates applied are 0.33 Mbit/s for video and 96 Kbit/s for audio.

4 Conclusion

This research concludes that the implementation of the video compression feature in the web-based Digital Knowledge Repository Application has been functionally successful, as demonstrated by the system's functional testing results. The effectiveness of the compression has also been validated by trial results, which show a storage savings of 47.42%. This effectiveness was achieved using the libvpx-vp9 encoder for video and libopus for audio, with a video bitrate of 0.33 Mbit/s and an audio bitrate of 96 Kbit/s.

References

El-Arsh, H. Y., Elliethy, A. S., Abdelaziz, A. M., & Aly, H. A. (2021). Performance comparison among popular implementations of H.264 encoders. *IOP Conference Series: Materials Science and Engineering*, *1172*(1), 012036. https://doi.org/10.1088/1757-899x/1172/1/012036

FFMPEG. (2010). *About FFmpeg*. Ffmpeg.Org. https://www.ffmpeg.org/about.html

Gupta, M., Shah, S., & Salmani, S. (2021). Improving whatsapp Video Statuses using FFMPEG and Software based encoding. *Proceedings - International Conference on Communication, Information and Computing Technology, ICCICT 2021*, *June 2021*.

https://doi.org/10.1109/ICCICT50803.2021.9510129

Kristiadi, D., & Marwiyati. (2021). Adaptive streaming server dengan FFMPEG dan Golang. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(3), 413–420. https://doi.org/10.29207/resti.v5i3.2998

Mao, B., Wu, S., Jiang, H., Yang, Y., & Xi, Z. (2018). EDC: Improving the performance and space efficiency of flash-based storage systems with elastic data compression. *IEEE Transactions on Parallel and Distributed Systems*, *29*(6), 1261–1274. https://doi.org/10.1109/TPDS.2018.2794966

Nugroho, M. R. (2021). Digital knowledge repository sebagai media pusat pengelolaan ilmu danpengetahuan. *Jurnal Teknologi Informasi Dan Komputer*, *7*(4), 464–472.

Pavlic, J., & Burkeljca, J. (2019). FFMPEG based coding efficiency comparison of H.264/AVC, H.265/HEVC and VP9 video coding standards for video hosting websites. *International Journal of Computer Applications*, *182*(37), 1–8. https://doi.org/10.5120/ijca2019918381

Safin, R., Garipova, E., Lavrenov, R., Li, H., Svinin, M., & Magid, E. (2020). Hardware and software video encoding comparison. *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2020*, *September*, 924–929. https://doi.org/10.23919/sice48898.2020.9240439

Smiti, P., Srivastava, S., & Rakesh, N. (2018). Video and audio streaming issues in multimedia application. *Proceedings of the 8th International Conference Confluence 2018 on Cloud Computing, Data Science and Engineering, Confluence 2018*, 360–365. https://doi.org/10.1109/CONFLUENCE.2018.8442823

Triantoro, A., Sajati, H., & Pujiastuti, A. (2018). The Audio Video of Web-Based Compression with FFMPEG. *Compiler*, *7*(2), 132. https://doi.org/10.28989/compiler.v7i2.280

Zeng, H., Zhang, Z., & Shi, L. (2017). Research and implementation of video codec based on FFMPEG. *Proceedings - 2016 International Conference on Network and Information Systems for Computers, ICNISC 2016*, 184–188. https://doi.org/10.1109/ICNISC.2016.54