Development of Audio Watermarking Web Services

Bervianto Leo Pratama
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
bervianto.leo@gmail.com

Rinaldi Munir
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
rinaldi.munir@itb.ac.id

***Abstract*—Internet has been used to share content as music, pictures, and videos. The contents are easy to get and share but hard to protect the copyright. A way to protect copyright of a content is digital watermarking, especially audio watermarking for audio content or music. There is some technique to do audio watermarking such as using Discrete Wavelet Transform (DWT). Audio watermarking as a software can become a component that integrate with web application. Audio watermarking component using DWT Based Algorithm. Integration of the component becomes audio watermarking web services. Development of Audio Watermarking Web Services based on Component Based Development. Audio watermarking web services passed the test, component testing for audio watermark component (robustness and imperceptibility) also system testing. The purpose of audio watermarking web services is to provide another developer to easy use and integrate it. The system and web services need to improve in security aspect.**

Keywords—audio watermarking, copyright, development, discrete wavelet transform, web services

I. Introduction

Internet becomes wide and popular for people to download content likes images, music and videos [1]. Internet is the best distribution system for digital media because cheap and almost instant. Internet has good impact but also have an issue, that is how to protect the media from piracy. There is a method to protect copyright for the content, it is digital watermarking.

There are many kinds of Digital Watermarking. Those are audio watermarking, image watermarking, and video watermarking. The kind of digital watermarking distinguished from the media that will be inserted by a watermark. Therefore, audio watermarking is a method to insert data into digital audio.

There some technique to insert data into digital audio, for example is using Least Significant Bit, Discrete Wavelet Transform, etc. The watermark for audio watermarking can be inserted into time domain and frequency domain. Based on domain that the watermarked inserted, the technique of audio watermarking can be categorized to Time-Domain Watermarking Techniques, Transform-Domain Watermarking Techniques, Compressed Domain Audio Watermarking, and Combined Domain Audio Watermarking [2].

Those are some implementation of Audio Watermarking that used server to do watermarking. Research by Pandit [3], Pandit build an Android Application that call the server to do watermarking. Another implementation that using a server to do watermarking in literature [4] and [5]. Purpose of building web service for audio watermarking is to provide developer build in easy to use audio watermarking so developer can only prepare the user interface.

II. Audio Watermarking

A. Aspect of Audio Watermarking

There is some aspect to evaluate the audio watermarking technique. The aspect are imperceptibility, robustness, security and data payload. Imperceptibility achieved when human can't distinguish audio that inserted with watermark with original audio [2]. Robustness achieved when watermark that inserted in audio not removed after some signal processing [2]. Security achieved when watermark survive from watermark changing or information changing in watermark, also the watermarking process must be safe from unauthorized parties. Bit rate or data payload means how much bit of watermark information can be save in signal each second.

B. Attack to Audio Watermarking

There is some attack to audio watermarking but not limited to this list [2],

1) *Additive and multiplicative noise*
2) *Linear and non-linear filtering*
3) *Data compression*
4) *Quantization from sample value*
5) *Temporal scaling*
6) *Delete or inserted sample*
7) *Averaging a double watermark of the signal.*
8) *D/A and A/D conversion*

III. Web Services

Web services is an interface that provided in network for bring functional access to another application. Web services provided system integration without depend on programming language and operating system [6]. Web services usually used and suit for integrate heterogenous system.

Advantage of using web services is the applications will loosely coupled and becomes reusable software component. Within that advantage, for business will decrease integration cost also complexity of software [6].

Web services have some type of communication such as SOAP (Simple Object Access Protocol) and REST (Representational State Transfer). RESTful application use HTTP request to post data, read data, and delete data. The data usually called resource.

A. Terms in REST Web Services

REST have three important terms that is resources, verb and representation. Resources is core element of web platform. Each resource must have unique identity usually known as URI (Universal Resource Identifier), example URL

(Uniform Resource Locator) for web address. Resources should be defined with noun not verb [7].

The verb is HTTP action such as POST, GET, PUT, DELETE, OPTIONS, etc [7]. Verb is action to modify or create a resource. As example, POST used to create new data in resource.

Representation is a media for showing resource to client. REST support some format like JSON and XML for represent the resource.

B. *Web Services Testing*

Web services need testing like another software. Testing for web services are functional testing, non-functional testing, and security testing [8]. Functional testing is about the software running well and input-output of web services as expected. Integration testing and system integration testing also can be done in functional testing.

Some important aspect in non-functional testing for web services as follows [8],

1. Concurrency is simultaneous access testing of services by many clients.
2. Throughput is the maximum number of transactions a service can be handled.
3. Payload is a number request that comes before the service behaves improperly.
4. Response time is the time required by the system to provide a response to a request.
5. Utilization is a resource used by the system when working.

According that aspect, the testing can be done as follows [8],

1. Load test to perform testing using some jobs and evaluate the amount that can be done and controlled by the system.
2. Stress test is testing the system to look at the limit of the system, as example how many errors that occur when doing a lot of work.
3. Endurance test is testing the system's ability to do work constantly. This test can detect the existence of bottleneck, memory leak, or decreased performance.
4. Spike test to evaluate system behavior when certain jobs are given. The purpose of this test is to determine when the system will fail or whether it can control drastic changes in a job.

The next test is testing the security of web services. This security test needs to pay attention to the following [8],

1. Authentication is checking method whether a request comes from an authorized source or is granted permissions.
2. Authorization is granting access to something so that it can access the service.
3. Penetration, testing of penetration is a simulated attack of something dangerous. This test looks for any loopholes that can result in system failure and stolen by someone who is not supposed to own the data.
4. Protocol or encryption standard testing, is a test that ensures every transaction is encrypted using certain defined techniques. Use of secure encryption standards may prevent an experiment to decrypt a channel or may be referred to as encryption attacks.

Three types of testing can be used to assess the ability of web services or failures that can occur in the web service. Security testing in the web service is necessary because difficult to control the client using web service. Information from the web service should ensure that any information can only be accessed by an authorized client.

IV. DEVELPOMENT TECHNIQUE

Development of Audio Watermarking Web Services using Component Based Development approach. First step for developing Audio Watermarking Web Services are determine the system and software requirements. Design of Web Services Architecture is important task.

Second step are find any component that corresponding of the requirement. Important task of this step is listing the component that fit within the requirement. The component that will be used depend on our requirements and the integration step. Integration step are third step.

Why need integration step? In third step, testing the component integration is important task. Sometimes it will be conflict each other. In this step, some component will be eliminated also sometimes give some options to create the system. After all, need to test again the integration of all component that forming a system usually called system testing.

System testing are fourth step. In here, some testing such as functional testing and or non-functional testing are important task. In this step, checking the input output of web services whether it is appropriate with expected output can be done for functional testing. Performance test likes memory used and time taken are used for non-functional test.

V. REQUIREMENT AND DESIGN OF WEB SERVICES

First step of development Audio Watermarking Web Services are to determine the system requirement and software requirement. Architecture of Audio Watermarking Web Services can be seen in Figure 1. Web Services not handle account management and storage management. Web Services use another component or service to handle account and storage. Web Services only checking the token whether it is valid or not.

A. *Flow of System*

System have two main process, there are embedding and extraction. Figure 2 show the embedding process and Figure 3 show the extract process. Another application that mean the application used web services and call it within HTTP. The process will run when the token is correct.

B. *Web Services Specification*

Web Services specification that defined here for another application know how to call the web services. Sometimes the specification will change after implementation of web services, but it is not a problem. Versioning becomes important aspect to handle the different of specification input output. Table 1 are the specification for embedding process.

Table 2 are the specification for extraction process. Web Services using task or job approach, so the answer of asking job will fast and the result will send later. Web services not use callback to send the result. The result of a job can be see with call web services again use GET. Table 3 are the specification for show any history of embedding process. Table 4 are the specification for show any history of extract process. Web services also can return only for an id of job. Table 5 for specific id of embedding process and Table 6 for specific id of extraction process.

Table 1 Specification for Sending Embed Job

URL	POST /api/v1/embed
Input	image: string audio: string key: string accessToken: string method_option: string
Output	(application/json) EmbedData: id: integer status: string method_option: string created_at: DateTime updated_at: DateTime image_input: string audio_input: string user_id: string audio_output: string result: string

Table 2 Specification for Sending Extract Job

URL	POST /api/v1/extract
Input	original_audio: string watermarked_audio_input: string key: string size: string accessToken: string method_option: string
Output	(application/json) ExtractData: id: integer method_option: string created_at: DateTime update_at: DateTime watermarked_audio_input: string audio_input: string user_id: string audio_output: string result: string

Table 3 Specification for Get Histories of Embed Job

URL	GET /api/v1/embed
Input	accessToken: string
Output	(application/json) list of EmbedData
Details	EmbedData: id: integer status: string method_option: string created_at: DateTime updated_at: DateTime image_input: string audio_input: string user_id: string audio_output: string result: string

Table 4 Specification for Get Histories of Extract Job

URL	GET /api/v1/extract
Input	accessToken: string
Output	(application/json) List of ExtractData
Details	ExtractData: id: integer method_option: string created_at: DateTime update_at: DateTime watermarked_audio_input: string original_audio_input: string user_id: string audio_output: string result: string

Table 5 Specification for Get Specific History of Embed Job

URL	GET /api/v1/embed/{data_id}
Input	accessToken: string
Output	(application/json) Object of Embed Data
Details	EmbedData: id: integer status: string method_option: string created_at: DateTime updated_at: DateTime image_input: string audio_input: string user_id: string audio_output: string result: string

Table 6 Specification for Get Specific History of Extract Job

URL	GET /api/v1/extract/{history_id}
Input	accessToken: string
Output	(application/json) Object of ExtractData
Details	ExtractData: id: integer method_option: string created_at: DateTime update_at: DateTime watermarked_audio_input: string original_audio_input: string user_id: string audio_output: string result: string

VI. AUDIO WATERMARKING TECHNIQUE

Audio watermarking that used in this web services are based on Discrete Wavelet Transform (DWT). The algorithm of audio watermarking for this web services are inspiration also combination from technique that Qi [9] and Al-Haj [10] proposed. The audio watermarking that used in this web services as non-blind watermarking because need information about original audio and the watermark (image size). Embedding process as follows,

1. Checking of sample width. Audio with 8-bit sample width will process image as greyscale so only have

one matrix with value 0-255. Audio with 16-bit and more will process image as RGB (Red Green Blue) so have three matrix will be processed.

2. The matrix of image should be transform with Arnold Transform. Do *n* times Arnold Transform, *n* are from the key. The key should be change into integer representation, as example each character change into integer than sum them. The rotation can limited to 10 to 30 times. Arnold transform will be applied into each matrix so for RGB, R matrix, G matrix and B matrix should have same transformation and the number of rotation. The result of this step named scrambled image.
3. Transform the scrambled image into one dimensions array or vector. For RGB, need append three arrays into one array, order of array is important so must same with extract process.
4. Pick sample arrays, for stereo can choose either left or right channel but still need same with extraction process.
5. DWT 2-Level for the sample arrays. This process will get A1, A2, D1, D2.
6. Inserted values from arrays in step 3 into D2. Insert the values with sum, newD2 = D2 + arrays.
7. Inverse DWT from new coefficient.
8. If stereo, combine the new channel sounds with another channel that not processed.

Extract process as follows,

1. DWT 2-Level for original audio and watermarked audio.
2. Get difference of two D2. Image = watermarkedD2 – originalD2. To improve the speed, only do after the size of image.
3. Transform from array into matrix. For 16-bit sample width need split array into 3 arrays before transform each array into matrix.
4. Anti-Arnold into each matrix using key. The rule must be same with embed process. If from 3 matrix, should combine them into one image.

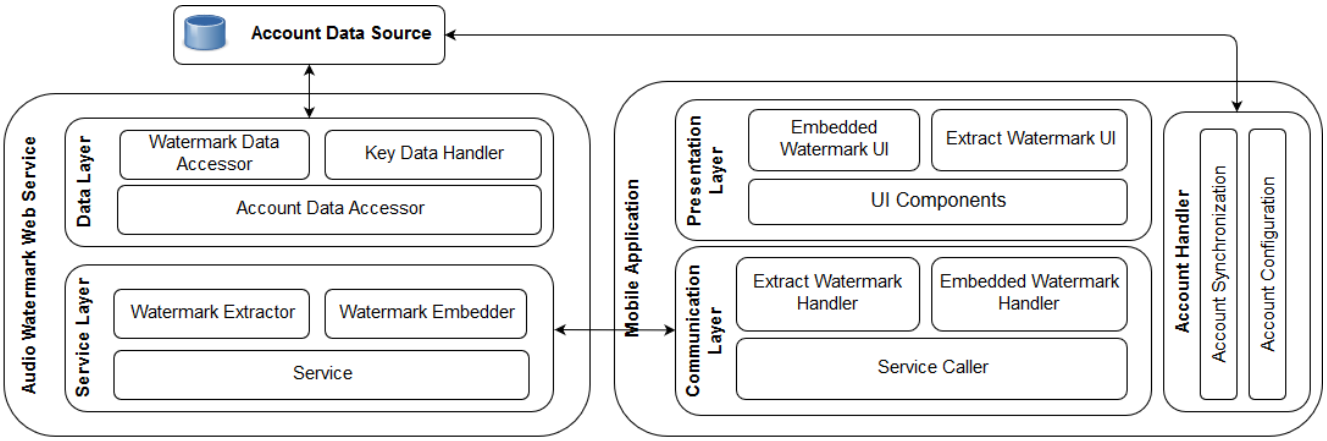


Figure 1 System Architecture

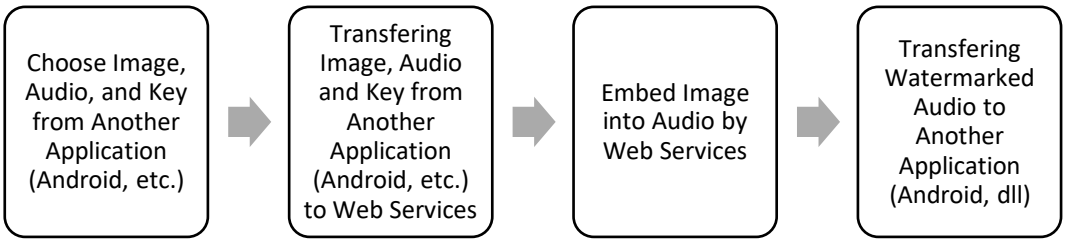


Figure 2 Embedding Process

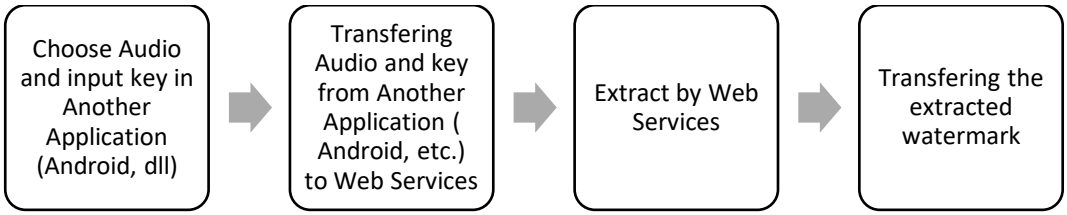


Figure 3 Extraction Process

VII. IMPLEMENTATION

The components that selected are PostgreSQL (Database), RabbitMQ (Task Caller), Celery (Worker), and use other services such as Firebase Storage for storage services also Firebase Authentication for authentication services. The Audio Watermarking Web Services wrapped into one services using Docker. Each component act as a service. Figure 4 show how the services communicate.

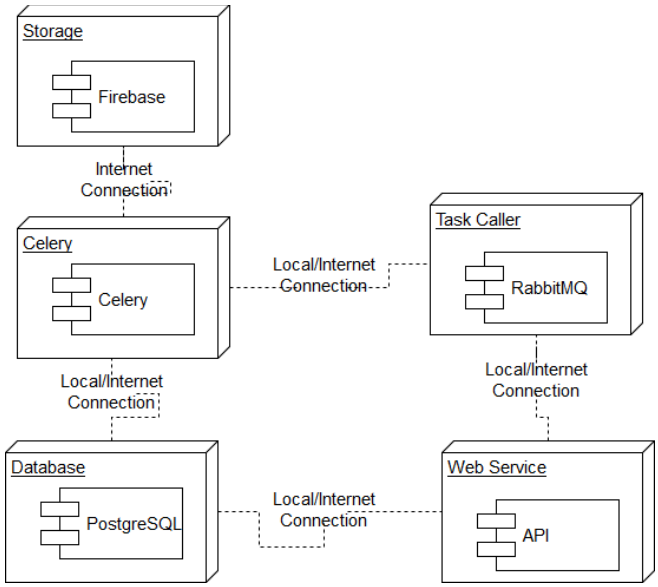


Figure 4 Deployment Diagram

Web services based on Python and using Django for web interfaces. Web Services component also develop using Component Based Development approach. Some library that used by the web services are,

1. OpenCV for image processing
2. PyWavelets for DWT and IDWT
3. Django REST Framework for plugin Django to used REST API
4. Firebase Admin for communication with Firebase Storage and Firebase Authentication service.

Figure 5 shows an example of web interface when called GET from browser the web services. Figure 6 shows an example of android application that called GET to have a result from embed job.

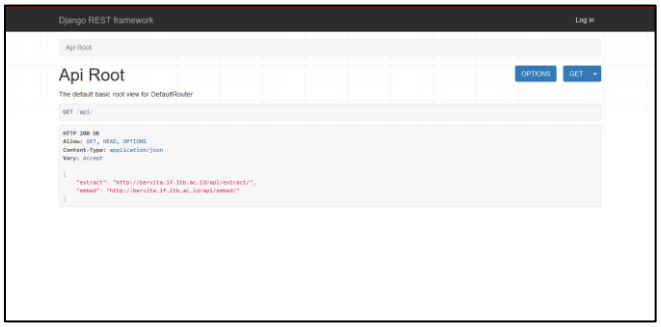


Figure 5 Main Interface of Web Services

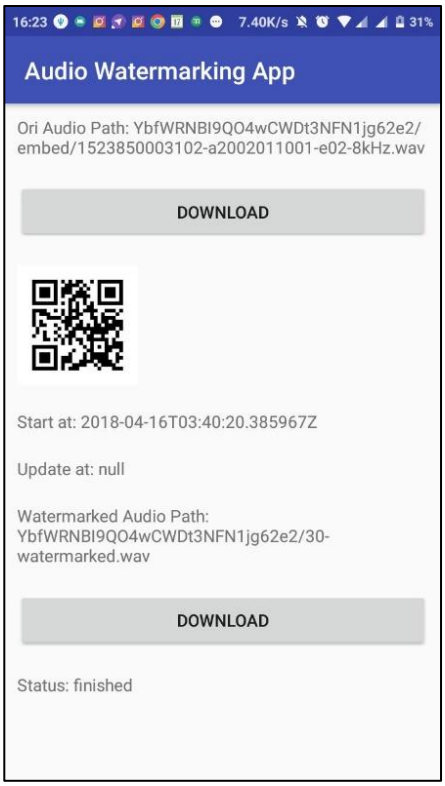


Figure 6 Example Interface that Show Embed Job Result

VIII. TESTING METHOD AND TESTING RESULT

There are three testing for test the audio watermarking web services. First, testing of functionality the web services. Second, testing of non-functionality the web services, performance test such as memory used by web services and time taken to process jobs. Third testing of audio watermarking performance. The environment of testing as Table 7.

Table 7 Hardware and Software Spesification

No	Subject	Spesification
1.	Operating System	Linux Ubuntu Server 16.04 LTS
2.	Memory	3 GB + 3 GB (swap)
3.	CPU	4 Core Intel Xeon E3-12xx v2 @ 2.5 GHz

A. *Functional Testing*

The main of functional testing are system testing. For each component must be well tested. Functional testing that applied to system testing are how the input-output with the expected output. Functional testing that performed have some level, from component testing, integration testing and system testing. All test case must be ensured to already passed.

B. *Non-Functional Testing*

Memory used by web services and other services (database, task caller and worker) when in same hardware not too big. Table 8 shows how many memory that used by web services. Table 9 and Table 10 shows time that apply when internet connection only to communicate with storage and authentication services. The times counted when job entered until job finished.

Table 8 Used Memory by The Services

No	Condition	Used Memory (MB)
1	*Idle*	352
2	Only GET called	441
3	1 job	546
4	2 jobs	560

Table 9 Elapsed Time for Doing Embed Job

No	Audio (MB)	Image (KB)	Elapsed Time (second)
1	1,3	2,1	5
2	3,4	2,1	6
3	4,6	2,1	9
4	17,6	2,1	9

Table 10 Elapsed Time for Doing Extract Job

No	Watermarked Audio (MB)	Original Audio (MB)	Elapsed Time (second)
1	1,3	1,3	4
2	3,4	3,4	4
3	4,6	4,6	5
4	17,6	17,6	6

C. Audio Watermarking Performance

Testing of Audio Watermarking performance pay attention to 4 aspects in Audio Watermarking, there are robustness, imperceptibility, security and data payload. Imperceptibility testing using questionnaire to 12 random people. Table 11 shows the result of questionnaire. Imperceptibility testing also can used visual method to see the wave differences.

Table 11 Imperceptibility Testing Result

Sound	Similarity		Regarded as the Original Audio	
	Similar	Difference	Original Audio	Watermarked Audio
8-bit	7	5	7	5
16-bit	5	7	7	5

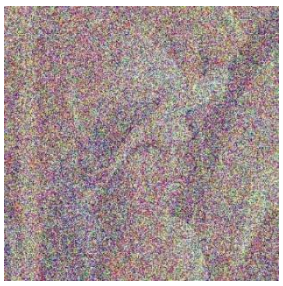
Robustness testing using some attack such as cutting, amplify and echo-attack. Figure 9 shows the difference of extracted image with the original watermark.

Security Testing based on image scrambling also the strength of key and image size that input from another application. Audio watermarking only have 21 keys different, so probability of key 1/21, it is not too small but good enough. Security improve with the probability of image size. Image size must exactly the size of original image, if not it will only show random image. Bruteforce attack must trying for 21 * width of image. That attack only works for registered user, so the web services can track any strange jobs. Anonymous user can't do the job.

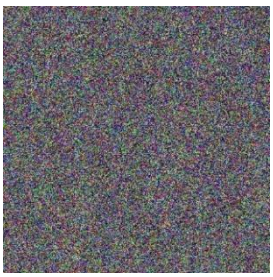
Main purpose of data payload testing is to know how much the watermark can embed into audio. Table 12 shows the result of data payload testing. Data payload depend on image dimension and sample size of audio. The relation of sample size and image dimension depend on sample width, the maximum dimension of 8-bit audio are $\sqrt{x}$, x are sample width size. $\sqrt{\frac{x}{3}}$ for 16-bit or more audio file.

Figure 7 Original Image in 16-bit Audio

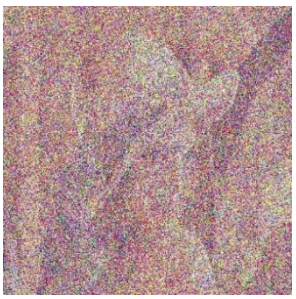
Figure 8 Original Image in 8-bit Audio



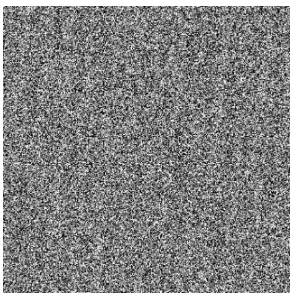
(a)



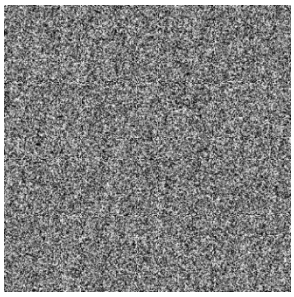
(b)



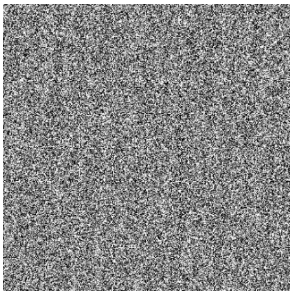
(c)



(d)



(e)



(f)

Figure 9 Extracted Watermark when (a) amplify, (b) cutting, (c) echo-attack on 16-bit audio, (d) amplify, (e) cutting, (f) echo-attack on 8-bit audio

Table 12 Data Payload Testing Result

No	Sample Width	Audio Size (MB)	Sample Size	Image Dimension (pixel)	Image Size
1	8-bit	4,56	2395137	1547 x 1547	16,2 KB
2	16-bit	17,6	4616551	1240 x 1240	4,39 MB

Based on the result of testing, the audio watermarking technique that used in this web services still achieve of 4 aspects. The technique not the best but feasible to use.

IX. Conslusion and Future Works

Development of Audio Watermarking Web Services using Component Based Development approach. Step by step of development of Audio Watermarking Web Services as follows,

1. Determine software requirement and the required component.
2. Find and choose the component that according to software requirements.
3. Integrate components and test.

Audio Watermarking Web Services that created in this project can be used by every application who using HTTP connection to call the services. For example, Android Application can call the web services within HTTP method also sending required data.

Audio Watermarking Technique that used in this final project need to be improved, such as for 8-bit audio need improvement to support greyscale also for 16-bit audio need to improve in imperceptibility aspect. The component or services of storage can change with another component that support local connection, so all component communicate in local connection, this will improve in delivery speed. Web services only have tested in functional requirement but still need security focus testing. This web services also can be improved with added another audio watermarking technique so can increased option to do audio watermarking.

Acknowledgment

I praise the Lord because I can finish my final project also this paper. Thanks to my lecture in final project, Dr. Ir. Rinaldi Munir, M.T., he taught me about watermarking and guide me to finish my final project. I also appreciate to Component Based Development's Lecture that brings some inspiration with new approach for software development.

References

[1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich and T. Kalker, Digital Watermarking and Steganography, Burlington: Morgan Kaufmann, 2008.

[2] S. P. S. Chauhan and S. A. M. Rizvi, "A survey: Digital Audio Watermarking Techniques and Applications," in *2013 4th International Conference on Computer and Communication Technology (ICCCT)*, Kuala Lumpur, 2013.

[3] M. K. Pandit and M. V. Bhosale, "Location Based Steganography On Mobile Smartphone Using Android Platform," *IJREAT International Journal of Research in Engineering & Advanced Technology*, pp. 127-130, 2015.

[4] L. Lovebbi and D. Z. Sudirman, "Rancang Bangun Aplikasi Steganografi dengan Metode Least Significant Bit di Audio pada Sistem Operasi Android," *ULTIMATICS Jurnal Teknik Informatika*, pp. 7-15, 2012

[5] L. Kumar and K. Sharma, "Web Based Novel Technique for Watermarking Colour Images on Android Mobile Phones," *International Journal of Advanced Research in Computer Science and Software Engineering*, pp. 47-52, 2013

[6] L. V. Bhandari and A. P. Wadhe, "Review Paper on Web Service Security," *International Journal on Computer Science and Engineering*, pp. 106-112, 2014.

[7] Infosys Limited, "Best Practices for Building RESTful Web Services," Infosys Limited, Bengaluru, 2017.

[8] H. James, "Web Services Testing," 8 January 2013. [Online]. Available: https://cdn.planittesting.com/planit/files/planit/57/5759be58-aed9-4737-8a77-f27027bffd43.pdf.

[9] W. Qi, X.-J. Chen and D. Xu, "A Novel Semi-Fragile Audio Watermarking Technique Based on Support Vector Regression," in *2015 Ninth International Conference on Frontier of Computer Science and Technology*, Dalian, 2015.

[10] A. Al-Haj, A. Mohammad and L. Bata, "DWT-Based Audio Watermarking," *The International Arab Journal of Information Technology*, pp. 326-333, 2011.