Developing Music Player Feature on October CMS Theme with RainLab's Blog Plugin

Atika Azzahra Akbar - 13514077

Informatics Engineering
School of Electrical Engineering and Informatics
Bandung Institute of Technology, Jl. Ganesha 10, Bandung 40132, Indonesia
13514077@stei.itb.ac.id

Abstract—**Blogging is a popular activity that is now considered as a way to make money by several people. With this reason, the activity of blogging nowadays also includes paying attention to looks and features of the blog. The looks and features are now a way to increase audience engagement. One of the feature that could increase user engagement is music player. Unfortunately, music player feature is usually provided as paid plugin. In this paper, a simple way to make music player feature is developed using HTML5 Audio, JQuery, and CSS on October CMS with Rain blog plugin. At the end of this paper, a fully functional simple audio player could be used just by writing an HTML5 Audio code on the post content of Rain blog plugin using Markdown Language.**

Keywords—audio; blog; CMS; music; player; plugin;

I. INTRODUCTION

For many years blogging has become one of the most popular activity done by people on the Internet. Even with the popularity of social media, there were still 2.7 million blog post published every day in 2016 alone [1]. Wordpress stated that their users produced about 84.3 million new posts each month [2]. A lot of people has made money from blogging and it is considered an occupation now. That is why, the activity of blogging is not all about writing posts anymore but also paying attention to the looks of the blog and the features. A good theme and useful features invites audiences [3]. One example of blogging feature that is good for audience engagement is the music player. Based on research about neurochemistry of music conducted by a team led by Prof. Daniel J. of McGill University's Psychology Department, it is proven that music reduces levels of stress and anxiety [4].

Due to the big demands of great blogging themes and features, a lot of easy to use tools have been created such as Blogger, Wordpress, or Wix. Unfortunately, these blogging platforms are not flexible to customize because they were made for people who have no technical ability. Meanwhile, the good plugins that are available with full support on these platform are usually not free, this also apply for the music player feature. With this reason, blogger with less budget has less chance of enhancing their blog and getting more audiences.

In this paper, we present a simple solution to develop a music player feature on October CMS theme and blog plugin by RainLab. The music player feature will be installed at the post page as a way to make the audience relax and comfortable while reading the blog. October CMS was chosen for this solution because it is a free, open source platform. It's easier to customize and developer has full control of its technologies to develop a theme or custom features.

II. LITETATURE STUDIES

A. Laravel Framework

The framework used to build October CMS is Laravel PHP Framework. It is a free, open source web framework build with PHP language created by Taylor Otwell [6]. It implements the Model-view-controller (MVC) pattern which stated by Microsoft [5] is an architectural pattern that divides application into three components. These components are models, object that implements the logic of an application that is usually used to store data from database, views, component that managed the user interface, and lastly controller, component that handles user input, works with the models and renders the user interface. Some of the key point features in Laravel are mentioned below [7]:

- Bundles, a modular packaging system with dedicated dependency. In Laravel 4, composer helps on managing the dependency of framework-agnostic and Laravel-specific packages available for download from packagist repository.

- Eloquent ROM (object-relation mapping), an implementation of active record pattern that enforce constraint on relationship between database objects. It presents database tables as classes with its instances connected to a single table rows.

- Query Builder, methods provided to build queries and caching query results programmatically.

- Routing, defines link between components, making it easier to manage routes for later changes.

- Restful controllers, component that handles logic behind http request from user.

- View, component that display user interface. The template use for view is *Blade* templating. It's a template that combines data model and view by caching it into PHP code to improve performance.

- Migrations, component that provides version control of database schemas, making it possible and easier to apply

database changes, simplifying the updating and deployment of application.

B. HTML 5 Audio

HTML5 audio is a new component to embed audio in documents [10]. Before HTML5, the only way to put an audio content was by using flash or a plugin [11]. With HTML5 audio methods, properties, and events, we could manipulate audio component using JQuery. Below are the list of methods, properties, and events that HTML5 have:

Methods	Description
Load()	Reload audio
Play()	Start audio
Pause()	Pause audio

Property	Description
Autoplay	Set audio to automatically start playing when loaded
Controller	Return the current media controller of the audio
Controls	Return display controls
currentSrc	Return the url of current audio
currentTime	Return the current playback position
Duration	Return the length of audio
Ended	Return the whether the audio has ended or not
Loop	Set the audio for looping
Src	Set or return the source of audio
Volume	Set or return the volume of audio

Events	Description
Abort	Fires when audio is aborted
Durationchange	Fires when duration of audio is changed
volumechange	Fires when volume of audio is changed

C. Markdown Markup Language

Markdown is a lightweight that convert text to HTML [8]. It is one of markup language using plain text formatting syntax. The idea of Markdown syntax, stated by John Gruber [9], the author of Markdown, is to make a readable document but still publishable on HTML without looking like it has been marked up with tags. This Markdown is used by RainLab's blog plugin. Below are some of Markdown's main elements notations:

Notations	Renders to
#, ##, ###, ####, #####, ######	<h1>, <h2>, <h3>, <h4>, <h5>, <h6>
<!-- -->	Comments, should not been seen
___ / --- / ***	<hr>
text	<strong>
text	<em>
~~text~~	<del>
>	<blockquotes>
*/-/+	(unordered list)
1., 2., 3., ..	(ordered list)
`text`	<code>
``` text ```	Syntax highlighting
| data | data | | ---- | ---- | | 1 | 2 |	<table>
[link](#)	link
	images

III. METHODS

A. Installing RainLab Blog Plugin on October CMS

Before building the audio player script, the first thing to do is to choose the theme and install RainLab blog plugin. Here are the steps to add plugin on October CMS dashboard:
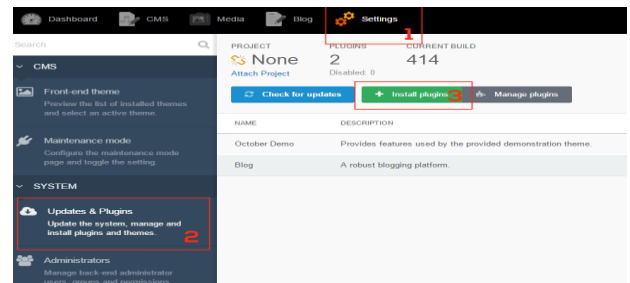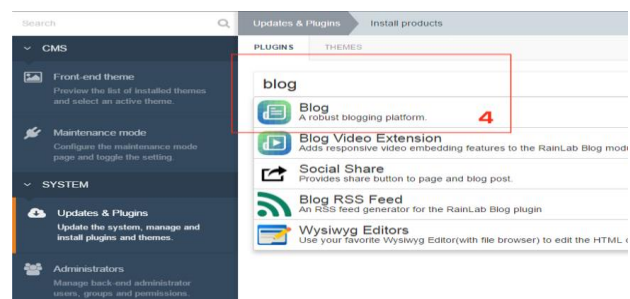

Figure 1 Adding blog plugin part 1

Figure 2 Adding blog plugin part 2

To download the plugin, click *settings* on dashboard menu. After that, choose the *Updates & Plugins* menu on the sidebar, follow by clicking the *install plugins* button. On the new page, search for the plugin called blog and click on the result.

Figure 3 Activate Theme

For choosing the theme on October CMS, choose the *Front End Theme* on the settings sidebar menu. On the new page, click the *Active* button next to the theme that will be activated. After the theme is activated, make an htm file to load the post generate by the Blog plugin. Here are the HTML code:

```
title = "Blog Post"
url = "/blog/:slug"
layout = "default"

[blogPost]
==
<?php
function onEnd()
{
    if (isset($this->post))
        $this->page->title = $this->post->title;
}
?>
==

{% if post %}
<div class="container post_container">
 <div class="posting">
    <div class="post_detail">
        <h2>{{ post.title }}</h2>
        <div class="post_content">
    {{ post.content_html|raw }}
        </div>
        <p class="info">
            Posted
            on {{ post.published_at|date('M d, Y') }}
        </p>
    </div>
 </div>
</div>
{% else %}
    <h2>Post not found</h2>
{% endif %}
```

B. Audio Player

First, make the HTML5 code for Audio player. There are three components in this audio player, they are play controller, stop/pause controller, and progress bar. Below is the HTML5 code and the looks of the audio player after some styling and image assets are added:

```
<div class="audio_player">
    <div id="play"></div>
    <div id="stop"></div>
    <div id='progressOut'>
        <div id='progressIn'></div>
    </div>
</div>
```


After the HTML code is made, link the components with the JQuery. With the script, the component play will be linked with audio play HTML5 events, the component pause will be linked with audio pause HTML5 events, and the progress bar will display the duration. In this script, the audio component is stored in HTML component with id *music* that will be retrieved by variable audioplayer. This *music* component isn't shown in the HTML code before because this component will be inputted by the user manually on the post using the RainLab blog plugin (this will be discussed in the next *subchapter*).

```
var audioplayer = document.getElementById('music');

if(audioplayer == null){
    $('.audio_player').hide();
} else {
    var playBtn = document.getElementById('play');
    var stopBtn = document.getElementById('stop');
    $('#stop').hide();
    $('#play').show();


    var playSound = function() {
     audioplayer.play();
     $('#play').hide();
     $('#stop').show();
    };

    var pauseSound = function() {
     audioplayer.pause()
     $(document).ready(function(){
        $('#stop').hide();
        $('#play').show();
     });
    };

    playBtn.addEventListener('click', playSound, false);
    stopBtn.addEventListener('click', pauseSound, false);

    audioplayer.addEventListener("timeupdate",
     updateProgress, false);
    function updateProgress() {

        var progress = $("#progressIn");
        var value = 0;

        if(audioplayer.duration == 'Infinity') { //If
    duration = infinity set value to 100
            value = 100;
        } else if (audioplayer.currentTime > 0) { //else if
    it is &amp;amp;amp;amp;amp;gt; 0 calculate percentage to
    highlight
            value = Math.floor((100 / audioplayer.duration) *
    audioplayer.currentTime);
        }
        progress.stop().animate({'width':value + '%'},500);
        audioplayer.addEventListener("timeupdate",
    updateProgress, false);
    }
}
```

C. Using Markdown Language to Input Music on RainLab Blog Plugin

To add the music for the audio player to play, just write an HTML code of audio component with the id *music* and link it to the source. October CMS, fortunately, has provided a folder to store media files called *media*. It can be accessed from the

dashboard menu. On the media page, clicked the upload button and upload the music file.
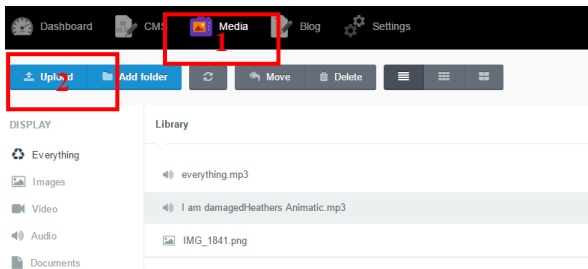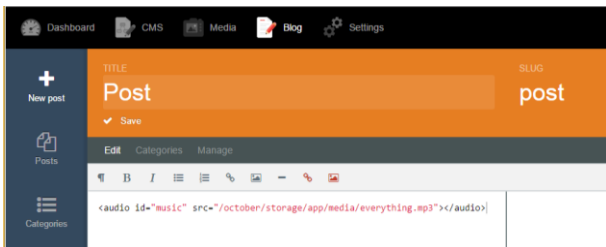

Figure 4 Add Media

Figure 5 Add HTML Audio on Post Content

In this case, the url of the music file was */october/storage/app/media/everything.mp3*. As mentioned before, write an HTML audio component with *music* as the id and put the url of the file inside the src. The code written inside the post content will be converted to HTML later because RainLab Blog plugin is using markdown language to create posts.

IV. RESULTS

After going through the steps in the chapter III methods, the audio player appears to be functioning well. The audio player has three functions which are play button, pause button, and progress bar. In contrast to other audio player plugin, this audio player uses less memory because it was developed with only one script. Other than that, unlike conventional audio player, to play the music the user doesn't have to open the file of the music. All it takes is to write an audio HTML5 component and include the music file's link to the source. Unfortunately, this audio player hasn't been developed to play more than one music.

Figure 6 Theme with the Music Player

V. CONCLUSIONS

To sum up, this paper provides a simple way to make a music player feature on October CMS Theme with RainLab Blog Plugin. The technologies used to develop this music player are Laravel PHP Framework, Markdown Language, HTML5 Audio, JQuery, and CSS. The steps are divided into three parts, setting the theme and plugin on October CMS, developing the audio player, and using Markdown language on Rain blog plugin as the music input. Even though a fully functional audio player is made, further work still needs to be done to enhance this audio player such as a function to play multiple music.

REFERENCES

[1] Internet Stats and Facts for 2016 [Web log post]. Retrieved May, 2, 2017, from https://hostingfacts.com/internet-facts-stats-2016/

[2] A live look at activity across Wordpress.com [Web log post]. Retrieved May, 2, 2017, from https://wordpress.com/activity

[3] Hill, Kristi. (2015, April 20) Is Your Wordpress Theme Sexy or Smart? [Web log post]. Retrieved May, 2, 2017, from http://blogambitions.com/importance-of-theme/

[4] (2013, March 27) *Major health benefits of music uncovered*. Retrieved May, 2, 2017, from https://www.mcgill.ca/

[5] ASP.NET MVC Overview. Retrieved May, 2, 2017, from https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx

[6] Taylor Otwell Profile. Retrieved May, 2, 2017, from https://laravel-news.com/@taylorotwell

[7] Bean, Martin (2015). *Laravel 5 Essential*. Mumbai: PACKT. Retrieved May, 2, 2017, from https://books.google.co.id/books?id=BWO4CAAAQBAJ&printsec=frontcover&redir_esc=y#v=onepage&q&f=false

[8] Gruber, John. (2004, December 17) Markdown Introduction. Retrieved May, 2, 2017 from https://daringfireball.net/projects/markdown/

[9] Markdown Syntax. Retrieved May, 2, 2017 from https://learn.getgrav.org/content/markdown

[10] Audio. Retrieved May, 2, 2017, from https://developer.mozilla.org/en/docs/Web/HTML/Element/audio

[11] HTML Audio/Video DOM Reference. Retrieved May, 2, 2017 from https://www.w3schools.com/tags/ref_av_dom.asp

Bandung, 6 May 2017

Atika Azzahra Akbar
13514077