KTF UI'S MASTER TEST PLAN

Created by [BigBang]
Issued March 29, 2016

Version History

V. No.	Description	Author	Date
1.0	Modified template as per group decision on the sections to include / exclude.	Anissa Lintang, Kega Kurniawan	2 April 2016
1.1	Modified template on tests targeted	Marvin Mitchell	3 April 2016
1.2	Modified template for scenario case per test	Anissa Lintang, Dion Edo	9 April 2016

Table of Contents

1. Introduction

The primary goal of this project is to develop a website for Komunitas Tari Fisip, Universitas Indonesia (KTF UI). (KTF UI) wishes to have a professional website that look like a company profile. The website shows the documentation on their show such as text, photos and videos linked to youtube. The site have approximately 10 pages and all of it have a responsive website template. This test plan contains a comprehensive list of tests that will be performed along with a workflow of how the tests will be executed.

1.1 Purpose

The purpose of the Iteration Test Plan is to gather all of the information necessary to plan and control the test effort for this phase.

This Test Plan for the KTF UI's Website supports the following objectives:

- Identify the requirements that are to be tested.
- Outline the testing approach that will be used.
- Describe the workflow of the testing process that must be executed.
- Provide a timeline with milestones for the testing phase.

1.2 Scope

This document is intended to provide a test plan to test the KTF UI's Website, which Bigbang developed. The test plan will consist of unit, integration, function, user interface, load, and configuration. Testing techniques that will be performed include white box and black box testing, and basis path testing. A test plan workflow will also be included along with milestones that have been set for this phase.

1.3 Document Terminology and Acronyms

Term	Definition
KTF UI	Komunitas Tari Fisip, Universitas Indonesia
UI	User Interface
BVA	Boundary Value Analysis
GUI	Graphical User Interface
API	Application Programming Interface

1.4 References

Ini di isi segala macem referensi buat projek ini

1.5 Document Structure

The remainder of this document is divided into following major parts: evaluation mission and test motivation, target test items, outline of planned tests, test approach and testing workflow, iteration milestones. The evaluation mission and test motivation contains a brief background on this project, its objectives and motivators for testing. The target test items and outline of planned tests include what will be tested and what tests will not be performed. The test approach contains the actual tests that were performed and how the tests were carried out. The testing workflow contains the workflow that BigBang Team followed in this phase. The last two sections contain the milestones of this phase.

2. Evaluation Mission and Test Motivation

The goal of this test plan is to ensure that the KTF UI's Website meets the specifications and design criteria of the two previous phases. Moreover, the test plan will provide a methodology on what the implementation team should test and the types of tests they will perform. Finally, the test plan will enable BigBang Team to release a stable KTF UI's Website.

2.1 Background

This project involves creating KTF UI's Website based on the requirements and design documents of the two previous phases. The website will be developed by the implementation using Wordpress CMS with template design and developed by BigBang Team. A comprehensive test plan has been developed to ensure that the website conforms to the specifications, design and to perform quality assurance on the final product. This will enable Bigbang Team release a complete and bug free KTF UI's Website and minimize the risk of website failure. The requirements document outlines the website's specifications and high-level requirements along with an analysis model with use cases diagrams of the website. The design document contains architectural and website interface, which is a foundation that the implementation team can create the website. The test plan will allow BigBang Team to verify if the final product successfully meets these specifications with a variety of testing techniques. The plan will also help in fault detection with the test cases that have been designed. The requirements and design documents are available at

https://github.com/gunadarma-academy/asde-big-bang.

2.2 Evaluation Mission

The three main objectives of this plan are:

- Ensuring that the specifications of the requirements document have been achieved.
- Ensuring that the specifications of the design document have been achieved.
- Ensuring that the risk of website failure is reduced to a minimum.

To achieve these objectives, BigBang Team has developed a test plan to verify that these objectives have been met. Meeting these objectives will enable BigBang Team to release a stable version KTF UI's Website.

2.3 Test Motivators

The targeted test items listed below will be the motivation for testing in this phase.

Unit Testing	A select number of methods will be tested in a couple of classes with black and white box testing to ensure that they function correctly

Function Testing	Will ensure that the use cases have been met
User Interface Testing	Will verify if the requirements of the GUI have been implemented as specified
Load and Stress Testing	See how the website performs when being access at its limits
Configuration Testing	Ensure that the website works correctly under different environment configurations
Security and Access Control Testing	Ensure and verify the website only can be accessed by user authorized

2.4 Test Milestone

Milestone of the master test plan will start from the 4th week due to week 1-3 is a phase of collecting user requirements, design and development of KTF UI's website.

Week	To-do Tests	Performer
4	Unit Testing	Marvin Mitchell. Dion Edo F
4	Function Testing	Anissa Lintang Dion Edo
4	Security and Access Control Testing	Kega Kurniawan
5	Load and Stress Testing	Marvin Mithcell Kega Kurniawan
5	Configuration Testing	Anissa Lintang
6	User Interface Testing	Dion Edo

3. Target Test Items

In this section, we will list the target test items. These are the items that should be tested. Due to time restrictions, we were not able to document and generate test cases for all the target test items; therefore, although we list all the target test items, we only provide a detailed test plan for a few of the major test items. For ease of reference, we have categorized the test items by motivation.

Unit Testing
Unit testing consists of testing all the different units of the system, in isolation. In essence, we must therefore test each class in isolation, and each method in isolation using white box and black box techniques. The list of test items for unit testing consists of all the classes and all their methods, as per the design document. Below is a list of the test items for which test cases have been generated and included in this document:

- Function pureal_init_style
- Function pureal_init_wp_setup
- Function pureal_sanitize_image
- Function pureal_init_widget
- Function pureal_set_excerpt_length
- Function pureal_init_theme_customization
- Function pureal_update_custom_css
- Function pureal_preview_js

Function Testing
Function testing consists of testing all the requirements and specifications from user, as per the requirements and specifications document. In essence, the list of functions to test corresponds to the list of use cases and requirements in the requirements document. Due to the importance of function testing, we have included detailed test cases for all the product functions. Below is the list of functions that were tested:

- Request profile
- Request information
- Request multimedia
- Insert content
- Edit and update content

User Interface Testing
User interface testing is concerned with making sure that each functionality concerning the user interface is works as per the requirements defined in the design document. Below is a list of the User Interface items that were tested:

- Page Home
- Page What we offer

- Page Achievements
- Page Gallery
- Page Project
- Page Contacts
- Page About

Load and Stress Testing

Load and Stress Testing is a part of website performance testing, concerned with testing the system beyond the limits it was designed for. In this type of test, we have focused mainly on testing the website when accessed beyond the limit. The testing will using Apache Jmeter software. Below are the test items that were identified:

- Throughput
- Deviation
- Latency
- Sample time

Configuration Testing:

Configuration testing is concerned with testing the website under different environment configurations. In this type of test, we have focused on testing the website under different website browser. Below is a list of the website browser will be tested:

- Mozilla Firefox
- Google Chrome
- Safari
- Internet Explorer

Security and Access Control Testing

Security and Access Control Testing will be performed as the website contain any data that can’t be accessed if the user is not authorized. Below is a list that will be tested:

- Admin Login

4. Outline of Planned Tests

BigBang Team will perform the following test: unit testing, integration testing, function testing, user interface testing, load and stress testing, configuration testing and security and access control testing. The following tests will not be performed: data and database integrity testing, business cycle testing,and integrity testing.

4.1 Outline of Test Inclusions

The following tests will be performed to test the KTF UI's Website.

No	Test Inclusions	Description
1	*Unit Testing*	Unit testing will be performed with black box and white box testing. Black box testing will include boundary value analysis and equivalence partitioning. White box testing will include basis path testing.
2	*Function Testing*	Function testing will ensure that the use cases have been implemented correctly by verifying if they are present in the website.
3	*User Interface Testing*	The pages of website will be tested by comparing the requirements in the design document and with the actual implementation of the website.
4	*Load and Stress Testing*	Load testing will see how the website performs when being accessed at its limits. This will be achieved by testing the website with the maximum visitor at peak hour.
5	*Configuration Testing*	Configuration testing is concerned with testing the application under different website browser the users may have.
6	*Security and Access Control Testing*	Security test of the website will be conducted to test the log-in system in administrator page to distinguish between guest and administrator. Because in the website environment there are 2 different user, guest and administrator, which have different roles. The guest only can see and request information and in the other hand administrator have many role such as input, delete, and

		update every information in the website, and adjust the configuration of the website.

4.2 Outline of Test Exclusions

Due to the nature of KTF UI's Website implementation, certain tests will be excluded, which are listed below.

No	Test Exclusions	Description
1	*Database and Database Integrity Testing*	We do not perform database and database integrity testing because the website made by using Wordpress CMS (Content Management System) and the database setting and it's integrity has well developed.
2	*Business Cycle Testing*	Business cycle testing is not applicable to KTF UI's Website because although the website showing it's company profile, it does not include any commercial and business things.
3	*Integrity Testing*	We do not perform integrity testing because the website made by using Wordpress CMS (Content Management System) and all the content's integrity has well developed.

5. Test Approach

The Test Approach describes the recommended strategy for designing and implementing the required tests. In this section, we will be describing the details of the tests that need to be performed for each target test item that was identified. These tests will be organized into the following sub-sections:

1. Unit Testing
2. Function Testing
3. User Interface Testing
4. Load and Stress Testing
5. Configuration Testing
6. Security and Access Control Testing

Moreover, for each of these test motivators, test cases will be described in detail. For each test case, we will provide a description of the test case, the inputs (or steps to reproduce) of the test case, and the outputs (the expected results) of the test case.

5.1 Unit Testing

Unit testing will test individual components along with their functions in isolation. This low level form of testing will include black box testing and white box testing. In black box testing, the function's boundaries will be tested to see if any errors occur there. White box testing will verify that all the paths in the function are correct through basis path testing.

5.1.1 Function pureal_init_style

Initiation of style.css file.

5.1.1.1 Black Box Testing

Test case 1 => style.css file existed

Test case 2 => style.css file not existed

Tester Name	Marvin Mitchell	
Test Date		
Test Case	1	2
Expected Results	The web styling will works	The web work will properly, but there is no styling

Actual Results		
Bug Found?		

5.1.1.2 White Box Testing

Basis Path Testing

```
function pureal_init_style(){
        wp_enqueue_style( 'style', get_stylesheet_uri() );
}
```

5.1.2 Function pureal_init_wp_setup

Initiation of wordpress related file and functions through wordpress hooks

5.1.2.1 Black Box Testing

Test case 1 => correct function call

Test case 2 => wrong

Tester Name	Marvin Mitchell	
Test Date		
Test Case	1	2
Expected Results	The function will return true	The server will throw an error
Actual Results		
Bug Found?		

5.1.2.2 White Box Testing

Basis Path Testing

```
function pureal_init_wp_setup(){
        /* Navigation Menus */
        register_nav_menus( array(
                'primary' => __( 'Primary Menu' ),
                'footer' => __( 'Footer Menu' )
        ) );
        /* add feature image support */
        add_theme_support('post-thumbnails');
        add_image_size('small-thumbnail', 180, 120, true); //hardcrop=true
        add_image_size('fullwidth-banner', 920, 210, array('left', 'top'));

        /* add post format support */
        add_theme_support('post-formats', array('aside', 'gallery', 'link'));
    }
    add_action('after_setup_theme', 'pureal_init_wp_setup');
```

5.1.3 Function pureal_sanitize_image

Filter images to be included in the post

5.1.3.1 Black Box Testing

Test case 1 => the file is not image

Test case 2 => the file is an unknown image

Test case 3 => the file is an image

Tester Name	Marvin Mitchell		
Test Date			
Test Case	1	2	3
Expected Results	The function will return false	The function will return false	The function will return true
Actual Results			
Bug Found?			

5.1.3.2 White Box Testing

Basis Path Testing

```
function pureal_sanitize_image( $image, $setting ) {
        /*
        * Array of valid image file types.
        * The array includes image mime types that are included in wp_get_mime_types()
        */
        $mimes = array(
                'jpg|jpeg|jpe' => 'image/jpeg',
                'gif'          => 'image/gif',
                'png'          => 'image/png',
                'bmp'          => 'image/bmp',
                'tif|tiff'  => 'image/tiff',
                'ico'          => 'image/x-icon'
        );
        // Return an array with file extension and mime_type.
        $file = wp_check_filetype( $image, $mimes );
        // If $image has a valid mime_type, return it; otherwise, return the default.
        return ( $file['ext'] ? $image : $setting->default );
  }
```

5.1.4 Function pureal_init_widget

Initialize widget location for wordpress

5.1.4.1 Black Box Testing

Test case 1 => wrong widget location

Test case 2 => wrong wordpress function call

Tester Name	Marvin Mitchell	
Test Date		
Test Case	1	2
Expected Results	The function will return false	The server will throws an error
Actual Results		
Bug Found?		

5.1.4.2 White Box Testing

Basis Path Testing

```
function pureal_init_widget(){
        register_sidebar(array(
                'name' => 'Sidebar',
                'id' => 'sidebar1',
                'before_widget' => '<div class="widget-item">',
                'after_widget' => '</div>',
                'before_title' => '<h4 class="widget-item-title">',
                'after_title' => '</h4>',
        ));
        register_sidebar(array(
                'name' => 'Footer Area 1',
                'id' => 'footer1'
        ));
        register_sidebar(array(
                'name' => 'Footer Area 2',
                'id' => 'footer2'
        ));
        register_sidebar(array(
                'name' => 'Footer Area 3',
                'id' => 'footer3'
        ));
        register_sidebar(array(
                'name' => 'Footer Area 4',
                'id' => 'footer4'
        ));
  }
```

5.1.5 Function pureal_set_excerpt_length

Edit paragraph excerpt length

5.1.5.1 Black Box Testing

Test case 1 => set the excerpt to be higher than 0

Test case 2 => set the excerpt to 0

Test case 3 => set the excerpt to be lower than 0

Tester Name	Marvin Mitchell		
Test Date			
Test Case	1	2	3
Expected Results	The paragraph will be trimmed to the specified excerpt length	The paragraph will never be shown	The server will throw an error
Actual Results			
Bug Found?			

5.1.5.2 White Box Testing

Basis Path Testing

```
function pureal_set_excerpt_length(){
        return 30;
}
```

5.1.6 Function pureal_init_theme_customization

Initialization of theme customizer API. Customizer API responsible for dynamically change the theme styling.

5.1.6.1 Black Box Testing

Test case 1 => wrong Customizer function call

Test case 2 =>call the right Customizer function call

Tester Name	Marvin Mitchell	
Test Date		
Test Case	1	2
Expected Results	The server will throw an error.	The styling of the theme will be able to change dynamically according to the user needs
Actual Results		

Bug Found?		

5.1.6.2 White Box Testing

Basis Path Testing

```
function pureal_init_theme_customization($wp_customize){
        /* Panel for header options */
        $wp_customize->add_panel('header_options', array(
                'title' => __('Header Options'),
                'priority' => 30
        ));

        /* Panel for footer options */
        $wp_customize->add_panel('footer_options', array(
                'title' => __('Footer Options'),
                'priority' => 40
        ));

        /* Section for header colors */
        $wp_customize->add_section('header_colors', array(
                'title' => __('Header Colors', 'pureal'),
                'panel' => 'header_options',
                'priority' => 2
        ));

        /* Section for header size */
        $wp_customize->add_section('header_sizes', array(
                'title' => __('Header Sizes', 'pureal'),
                'panel' => 'header_options',
                'priority' => 3
        ));

        /* Header Link Color */
        $wp_customize->add_setting('header_link_color', array(
                'default' => '#fff',
                'transport' => 'postMessage'
        ));
           $wp_customize->add_control(new    WP_Customize_Color_Control($wp_customize,
'header_color_control', array(
                'label' => __('Header Link Color', 'pureal'),
                'section' => 'header_colors',
                'settings' => 'header_link_color'
        )) );

        ......

        /* Footer Link Hover Color and opacity */
        $wp_customize->add_setting('footer_link_hover_color', array(
                'default' => '#fff',
                'transport' => 'postMessage'
        ));
            $wp_customize->add_control(new    WP_Customize_Color_Control($wp_customize,
'footer_link_hover_control', array(
                'label' => __('Footer Link Hover Color', 'pureal'),
                'section' => 'footer_colors',
                'settings' => 'footer_link_hover_color'
        )) );
        $wp_customize->add_setting('footer_link_hover_opacity', array(
                'default' => 70,
                'transport' => 'postMessage'
        ));
        $wp_customize->add_control('footer_link_hover_opacity_control', array(
                'label' => __('Footer Link Hover Opacity', 'pureal'),
                'type' => 'range',
                'section' => 'footer_colors',
                'settings' => 'footer_link_hover_opacity'
        ));
    }
```

5.1.7 Function pureal_update_custom_css

The function needed to update style.css dynamically

5.1.7.1 Black Box Testing

Test case 1 => wrong function call

Test case 2 => Wrong css property

Test case 3 => right function call

Tester Name	Marvin Mitchell

Test Date			
Test Case	1	2	3
Expected Results	The server will throw an error	The web styling will behave weird.	The style.css file will be updated accordingly.
Actual Results			
Bug Found?			

5.1.7.2 White Box Testing

Basis Path Testing

```
function pureal_update_custom_css(){
?>
   <style type="text/css">
         #header-navigation{
                  padding: <?php echo '0 '.(get_theme_mod('header_pad_left_right', 10)*0.25).'%'; ?>;
                  background-color: <?php echo get_theme_mod('header_bg_color', '#4c1b1b'); ?>;
                  color: <?php echo get_theme_mod('header_text_color', '#fff'); ?>;
         }


         ........

         @media (max-width: 768px) {
                  /* General */
                  .row-container{
                           padding: 1% 5%;
                  }

                  .row, .col-12, .col-11, .col-10, .col-9, .col-8, .col-7, .col-6, .col-5, .col-4, .col-3, .col-2, .col-1{
                           display: block;
                           width: initial;
                  }
         }
   </style>
<?php
  }
```

5.1.8 Function pureal_preview_js

The javascript needed for Customizer API to dynamically change the theme styling

5.1.8.1 Black Box Testing

Test case 1 => wrong function call

Test case 2 => include the right javascript file

Tester Name	Marvin Mitchell	
Test Date		
Test Case	1	2
Expected Results	The server will throw an error	The web styling will change dynamically according to the user needs.
Actual Results		
Bug Found?		

5.1.8.2 White Box Testing

Basis Path Testing

```
function pureal_preview_js() {
        wp_enqueue_script('custom_css_preview',get_template_directory_uri().'/js
        /customizepreview.js', array( 'jquery','customize-preview' ) );
}
```

5.1.9 Function yScrollHandler

The javascript function to trigger css3 animation when scrolling occur.

5.1.9.1 Black Box Testing

Test case 1 => animations array is empty

Test case 2 => animations array length is greater than 0

Test case 2.1 => window scroll offset is bigger than the animation height trigger

Test case 2.2 => window scroll offset is smaller than the animation height trigger

Tester Name	Marvin Mitchell			
Test Date				
Test Case	1	2	2.1	2.2
Expected Results	The window scroll event will disabled	The scroll offset will be checked againt all animation height trigger	Animation will occur	Animation will not occur
Actual Results				
Bug Found?				

5.1.9.2 White Box Testing

Basis Path Testing

```
function yScrollHandler(){
   if(index < animations.length){
      if(window.pageYOffset > animations[index].trigger){
         $(animations[index].animTarget).addClass(animations[index].animation);
         ++index;
      }//end inner if
   }//end outerif
   else{
      window.onscroll = null;
   }
}
```

5.2 Function Testing

This section is concerned with testing the functions (or requirements) of the software. This is a critical aspect of the testing effort, as it ensures that the software meets the requirements, and thus ensures acceptance by the users. Despite time restrictions, we have included all the major product functions as well as test cases for each one.The following sections are devoted to the major functions that were selected as testing targets. Each section lists and describes the different test cases that are important to check.

Test Case 1	Request profile

Tester Name	Anissa Lintang
Test Date	
Test Case Description	User can request and access (see) profile from the website
Input (Steps)	1. Open the website 2. Select and Open the profile page
Expected Results	The website will shows the profile that the user access
Actual Results	

Test Case 2	Request information
Tester Name	Anissa Lintang
Test Date	
Test Case Description	User can request and access (see) information from the website
Input (Steps)	1. Open the website 2. Select and Open the page contained information the user want to know
Expected Results	The website will shows the information that the user access
Actual Results	

Test Case 3	Request multimedia
Tester Name	Anissa Lintang
Test Date	
Test Case Description	User can request and access (see) multimedia from the website
Input (Steps)	1. Open the website 2. Select and Open the page contained multimedia
Expected Results	The website will shows and play (if video) the multimedia that the user access
Actual Results	

Test Case 4	Insert content
Tester Name	Anissa Lintang
Test Date	
Test Case Description	Admin can insert new content to the website
Input (Steps)	1. Open the website 2. Admin login to dashboard 3. Admin select new content type 4. Insert the content
Expected Results	The website will have a new content that corresponding to the admin content inserted
Actual Results	

Test Case 5	Edit and Update content
Tester Name	Anissa Lintang
Test Date	
Test Case Description	Admin can edit and update existing content to the website
Input (Steps)	1. Open the website 2. Admin login to dashboard 3. Admin select existing content that want to edit and update 4. Edit and update the content
Expected Results	The website will have a content that edited and updated corresponding to the admin content
Actual Results	

5.3 User Interface Testing

To test the User Interface, each functionality described in the design document will be verified to see if it has been implemented correctly, if it responds normally and also if no errors occur during the process between the user and the game.

A schema will be used to test (unit), what is the purpose of the test (what is tested), what are the inputs (from the user for instance), what is the expected result and also what is the effective (real result).

Dion

Test Case 1	Test Slideshow
Tester Name	Dion Edo
Test Date	
Test Case Description	User can click three bullets in order to change the big profile picture
Input (Steps)	3. Open the website 4. Click Bullets Button in slideshow
Expected Results	The website will shows the picture multimedia that the user clicked
Actual Results	

Test Case 3	Check Company Profile Button
Tester Name	Dion Edo
Test Date	
Test Case Description	User can request and access (see) the whole Company Profle description in Company Profile Page
Input (Steps)	5. Open the website 6. Select and Open the page contained multimedia
Expected Results	The website will change directly into Company Profile Page
Actual Results	

Test Case 3	Check Social Media Access Link

Tester Name	Dion Edo
Test Date	
Test Case Description	User can request and access (see) Social Media of KTF UI
Input (Steps)	7. Open the website 8. Select and Open the page contained multimedia
Expected Results	The user bring into social media which button clicked (Twitter, FB, Youtube and Instagram)
Actual Results	

Test Case 3	Check Menu Navigation Access Link
Tester Name	Dion Edo
Test Date	
Test Case Description	User can request and access (see) each page of the menu navigation (Achievements, Project, Gallery, Contact)
Input (Steps)	9. Open the website 10. Select and click each menu navigation on right corner of the site
Expected Results	The website will shows each menu navigation page
Actual Results	

Test Case 3	
Tester Name	Dion Edo F
Test Date	
Test Case Description	
Input (Steps)	
Expected Results	
Actual Results	

There are many testing dont mention yet because website dont build complete yet.

List Testing:
Integrity Data: Data such as website profile description text fulfill the requirement and expected.
Button, Link and navigation availabe in another page which dont build complete yet
Font: Font has fulfill the expected
Color: Color has fulfill the expected

5.4 Load and Stress Testing

Load testing is normally concerned with testing the system beyond the limits it was designed for. However, due to restrictions we have placed on the number of visitor (unknown number) that can access in one time, we cannot, for instance, test to see if the website works with maximum visitor (undefinitely yet). Therefore, we will be testing the website as close as possible to the limits it was designed for. In fact, we will attempt to simulate a fully loaded board, where all the function are displayed. In this scenario, we will re-evaluate the website's functionalities and response times.

Kega

Test Case 1	Continues user visit
Tester Name	Kega Kurniawan
Test Date	
Test Description	On this test the website will be tested with the continues visit from multiple user in every 5 second for 30 minutes
Expected Results	Error result from the website can be minimize. Standard deviation and latency from result test is not to high
Actual Results	

Test Case 2	Many user visit the website in 1 time
Tester Name	Kega Kurniawan
Test Date	
Test Description	Access the websites with many user (the number of user will be described in scenario test later) in 1 time

Expected Results	Error result from the website can be maximize and standard deviation is high
Actual Results	

Test Case 3	Maximum user visit
Tester Name	Kega Kurniawan
Test Date	
Test Description	Try to reach maximum user of website can handle from increasing user visit in sequent time
Expected Results	The website is no longer can handle user request
Actual Results	

5.5 Configuration Testing

Configuration testing is concerned with testing the application under different environment configurations the users may have. For the KTF UI Website, we will be focusing on testing the website under different browser. As per the requirements document, this includes Mozilla Firefox, Google Chrome, IE and Safari. In order to simulate these different client environments, we will be using well-known different browser directly. In order to test the KTF UI Website under several different browser.

Test Case 1	Access website via Mozilla Firefox
Tester Name	Anissa Lintang
Test Date	
Test Description	Test and see if the website can be accessed and work properly via browser Mozilla Firefox
Expected Results	Website will be successfully accessed and work properly
Actual Results	

Test Case 2	Access website via Google Chrome
Tester Name	Anissa Lintang
Test Date	
Test Description	Test and see if the website can be accessed and work properly via browser Google Chrome
Expected Results	Website will be successfully accessed and work properly
Actual Results	

Test Case 3	Access website via Safari
Tester Name	Anissa Lintang
Test Date	
Test Description	Test and see if the website can be accessed and work properly via browser Safari
Expected Results	Website will be successfully accessed and work properly
Actual Results	

Test Case 4	Access website via Internet Explorer
Tester Name	Anissa Lintang
Test Date	
Test Description	Test and see if the website can be accessed and work properly via browser Internet Explorer
Expected Results	Website will be successfully accessed and work properly
Actual Results	

5.6 Security and Access Control Testing

Security and access control testing have to be conducted to ensure security system especially security for the admin system, it is necessary because this testing will be test the website how well can handle and prevent unauthorized user to access the admin system. In this testing we will conduct security system test for admin login, and see how well security system from admin login to prevent unauthorized user to enter the admin system. Because security and access system in this website has been already setup by CMS Wordpress, then we will only do black box testing for this testing.

Test Case 1	Login admin system using wrong username and wrong password
Tester Name	Kega Kurniawan
Expected Results	Login will failed
Actual Results	

Test Case 2	Login admin system using correct username and wrong password
Tester Name	Kega Kurniawan
Expected Results	Login will failed
Actual Results	

Test Case 3	Login admin system using wrong username and correct password
Tester Name	Kega Kurniawan
Expected Results	Login will failed
Actual Results	

Test Case 4	Login admin system using correct username and correct password
Tester Name	Kega Kurniawan
Expected Results	Login will succeed
Actual Results	